

Universidad Politécnica de Cartagena

Proyecto Fin de Grado

Desarrollo de una aplicación web RESTfull basada en
Spring y Angular

Jorge Zapata Alburquerque

Tutor: Juan Carlos Arnoutse Sánchez

*“No te rindas, por favor, no cedas,
aunque el frío queme,
aunque el miedo muerda,
aunque el sol se esconda y se calle el viento,
aún hay fuego en tu alma,
aún hay vida en tus sueños.”*

Mario Benedetti.

AGRADECIMIENTOS

A mis amigos por apoyarme, por compartir momentos de agobio y de frustración y saber siempre sacarme el lado bueno, o una sonrisa.

Especialmente a Lorena por soportarme durante la cuarentena por video llamada mientras terminaba la implementación.

A mis compañeros de piso, porque de una forma u otra han formado parte de esta andadura.

A mis compañeros de carrera y sobre todo a Carmen y a Pedro por haber estado siempre ahí cuando he necesitado algo, tanto dentro de la universidad, como fuera. Se que vais a conseguir todo lo que os propongáis, porque lo valéis.

A mi empresa Capgemini por darme la oportunidad de este TFG y además ofrecerme una oportunidad laboral.

A mi familia y sobre todo a mis padres creer en mí y no desesperarse, sin vuestro apoyo no habría logrado llegar hasta aquí.

Por último, a mi pareja Marta, que a pesar de haber llegado hace poco, es la que más me ha aguantado y ha apoyado con esto.

Muchas gracias por formar parte de esta historia.

Resumen

La motivación de este proyecto reside en la creación y desarrollo de una aplicación web de gestión de currículums dentro de la empresa Capgemini en la sede de la Región de Murcia. Esta empresa está dedicada a la consultoría, servicios tecnológicos y transformación digital, y debido a que actualmente el centro de Murcia está en crecimiento de proyectos por la llegada de la nueva tecnología del 5G, se requiere de una aplicación que ayude a esta empresa a gestionar de manera más rápida y eficaz los currículums de sus trabajadores para su posterior asignación en los diversos proyectos.

Este proyecto recibe el nombre de MGCV, y su uso está destinado al equipo de Recursos Humanos de Capgemini, con el objetivo de facilitar la búsqueda de técnicos para satisfacer las necesidades de nuevos proyectos, o proyectos ya existentes que sufran cambios.

Como ayuda para elaborar este proyecto se han utilizado algunas de las tecnologías más punteras en la actualidad, como Angular, Nodejs, Java, Spring Boot, o Jenkins.

Abstract

The motivation of this project lies in the creation and development of a web application for managing curricula within the company Capgemini in the headquarters of the Region of Murcia. This company is dedicated to consulting, technology services and digital transformation, and since the center of Murcia is currently a high demand in projects due to the arrival of new technology 5G, an application is required to help the company manage more quickly and effectively the resumes of its workers for their subsequent assignment in the various projects.

This project is called MGCV, and its use is intended for Capgemini's Human Resources team in order to facilitate the search for technicians to meet the needs of new projects or existing projects that undergo changes.

Some of today's most advanced technologies, such as Angular, Nodejs, Java, Spring Boot, and Jenkins, have been used to assist in the development of this project.

Índice

Capítulo 1 - Introducción.	14
1.1. Objetivos.	15
1.2. Aportaciones al ámbito empresarial.	15
1.3. Estructura del documento.	16
Capítulo 2 - El Estado del Arte	17
2.1. API REST o API RESTFull	18
2.1.1. Web Services	18
2.1.2. API	18
2.1.3. SOAP vs REST	18
2.2. Tecnologías Backend.	22
2.2.1. Java vs C#	23
2.2.2. Spring vs Django.	25
2.2.3. Maven vs Gradle.	27
2.2.4. PostgreSQL vs MySQL	29
2.3. Tecnologías Frontend.	31
2.3.1. Angular vs React.	32
Capítulo 3 – Recursos utilizados	34
3.1. Software.	34
3.1.1. Eclipse	34
3.1.2. Visual Studio Code	34
3.1.3. Node.js	35
3.1.4. pgAdmin	35
3.2. Librerías y Plugins	35
Capítulo 4 - Desarrollo del proyecto	36
4.1. Análisis	36
4.1.1. Requisitos del sistema	36
4.1.2. Requisitos del software	36
4.2. Diseño	38
4.2.1. Arquitectura	38
4.2.2. Base de Datos	39
4.2.3. Diseño de las interfaces de usuario	41
4.2.4. Diagrama de clases	44
4.3. Metodología de trabajo	45
4.3.1. Kanban	45
4.3.2. Git	45

4.4. Implementación	46
4.4.1. Servidor	46
4.4.2. Cliente	52
Componentes.	52
Estructura.	55
4.4.3. GitHub	56
4.4.4. Jenkins	57
Capítulo 5 - Conclusiones y trabajo futuro	58
5.1. Conclusiones	58
5.2. Trabajo futuro	59
Capítulo 6 – ANEXOS	60
ANEXO I: Manual de usuario.	60
ANEXO II: Manual de administrador.	66
ANEXO III: Instalación y configuración de Maven para Eclipse.	69
ANEXO IV: Bibliografía	79

Índice de Figuras

Figura 1: Backend.....	22
Figura 2: Logo de Java	23
Figura 3: Logo de C#.....	24
Figura 4: Logo de Spring.....	25
Figura 5: Logo de Django	26
Figura 6: Logo de Maven.....	28
Figura 7: Logo de Gradle	28
Figura 8: Logo de PostgreSQL	30
Figura 9: Logo de MySQL	30
Figura 10: Algunas tecnologías Frontend.....	31
Figura 11: Logo de Angular	32
Figura 12: Logo de React.....	33
Figura 13: Logo de Eclipse.....	34
Figura 14: Logo de Visual Studio Code.....	34
Figura 15: Logo de NodeJS	35
Figura 16: Logo de pgAdmin	35
Figura 17: Ilustración de las funcionalidades del servicio que tiene un usuario con rol "user"	37
Figura 18: Ilustración de las funcionalidades del servicio que tiene un usuario con rol "admin"	38
Figura 19: Ilustración del flujo de información a través de los dispositivos.	38
Figura 20: Estructura de la base de datos.....	40
Figura 21: Vista principal de la aplicación del perfil "user"	41
Figura 22: Pestaña "Curriculum".....	41
Figura 23: Pestaña "Curriculum-Skill"	42
Figura 24: Vista principal de la aplicación del perfil "admin"	42
Figura 25: Pestaña "School".....	43
Figura 26: Pestaña "Experiences"	43
Figura 27: Modelo Entidad - Relación de la aplicación.	44
Figura 28: Tablero de Trello de la aplicación	45
Figura 29: Eclipse Marketplace	46
Figura 30: Inicio de un proyecto Spring	47
Figura 31: Configuración de un proyecto Spring	48
Figura 32: Archivo pom.xml.....	49
Figura 33: Arquitectura del Backend	50
Figura 34: Ejemplo de una clase controller.....	51
Figura 35: Estructura de la carpeta generada "animals"	52
Figura 36	53
Figura 37: Interpretación de la figura 4.20 en un navegador	53
Figura 38: Estructura de un archivo CSS	54
Figura 39: Interpretación de la figura 4.22 en un navegador	54
Figura 40: Estructura del frontend.....	55
Figura 41: Ilustración que muestra la pantalla de login.....	60
Figura 42: Ilustración que muestra la página de inicio	60
Figura 43: Ilustración del inicio del formulario de registro de curriculum	61
Figura 44: Ilustración que muestra la pestaña de datos personales del formulario de registro de curriculum.....	61
Figura 45: Ilustración de la pestaña "añadir educación" del formulario de registro de curriculum ..	62

Figura 46: Ilustración de la pestaña “añadir experiencia” del formulario de registro de curriculum	62
Figura 47: Ilustración de la pestaña “añadir skill” del formulario de registro de curriculum	63
Figura 48: Ilustración del resumen de un curriculum	64
Figura 49: Ilustración que representa el resumen de todos los empleados con las skills que posee cada uno.....	65
Figura 50: Ilustración que representa el filtrado de la Skill “Angular 7”	65
Figura 51: Ilustración que muestra la página de inicio de la app.	66
Figura 52: Ilustración que muestra las Educaciones que hay creadas en la aplicación	66
Figura 53: Ilustración que muestra todas las universidades añadidas en la aplicación.....	67
Figura 54: Ilustración que muestra el desplegable de Grados disponibles para añadirlos o modificarlos en una universidad.....	68
Figura 55: Ilustración que representa el contenido “Este equipo”	69
Figura 56: ilustración que representa el contenido de “Sistema”.	70
Figura 57: Figura que representa el cuadro de “Configuración avanzada”	71
Figura 58: Ilustración que representa la pantalla de variables de entorno	72
Figura 59: Ilustración que representa el cuadro de nueva variable del sistema.	72
Figura 60: Ilustración que representa la pantalla de variables de entorno.	73
Figura 61: Ilustración que representa la pantalla de edición de una variable de entorno	74
Figura 62: Ilustración de la Consola de Windows	75
Figura 63: Ilustración que representa donde hay que acceder para instalar un “new software” en Eclipse	75
Figura 64: Ilustración que representa los sitios que tiene Eclipse disponibles para buscar software en ellos.	76
Figura 65: Ilustración que representa una lista de softwares instalados y disponibles para instalar.77	77
Figura 66: Ilustración que representa los pasos para crear un proyecto Maven	78

Capítulo 1 - Introducción.

En la actualidad nuestra sociedad está siendo testigo de la nueva era de la conectividad inteligente. La llegada del 5G hará que nuestra forma de ver la vida cambie por completo. Entre otras cosas se mejorará la conectividad y se disminuirá la latencia, se posibilitará que objetos cotidianos como nuestro coche o el alumbrado de la ciudad estén conectados entre sí, y además se multiplicará el número de dispositivos conectados en general, por lo que también aumentará el flujo de la información. Esto hace que cada vez más empresas quieran adaptarse y actualizarse a la Industria 4.0 y no quedarse atrás en esta nueva carrera tecnológica, pudiendo así también dotar de mayor conectividad a su negocio.

Capgemini^[1] es una empresa de origen francés que fue fundada en 1967 en la ciudad de Grenoble, en el sureste del país, y actualmente se establece como una de las empresas líder a nivel mundial en consultoría, servicios de tecnología y transformación digital. Además, dispone de más de 250 sedes repartidas en 44 países y cuenta con más de 200.000 empleados.

Particularizando este contexto en la Región de Murcia, la sede local de Capgemini en la Región se ha visto especialmente afectada por la llegada de esta nueva tecnología, experimentado un aumento de la demanda de proyectos y en consecuencia un respectivo aumento en su número de plantilla para que pueda cubrir esta nueva demanda. Esto ha generado ciertos problemas a la hora de poder coordinar y satisfacer las necesidades tanto de los proyectos activos, como de los nuevos que están por comenzar; por lo que se ha generado también una necesidad de poder agilizar y facilitar una clasificación más detallada de sus trabajadores, para así poder llevar a cabo una asignación de proyectos más eficaz y precisa.

Este TFG presenta una solución a esta necesidad que consiste en la creación de una aplicación que sea capaz de almacenar los currículums de los trabajadores para poder clasificarlos por proyectos según las skills que tenga cada uno de ellos, entendiendo por skills los conocimientos y/o habilidades que posean los mismos.

1.1. Objetivos.

El objetivo general de este proyecto es la creación de una aplicación que permita mantener un registro de las habilidades o skills que posee cada uno de los empleados que hay en la empresa Capgemini para su futura asignación en sus diversos proyectos.

Para ello se ha elaborado una serie de objetivos tecnológicos:

- Programación de una aplicación REST Full con ayuda de uno de los frameworks más utilizados de Java.
- Utilizar una base de datos relacional, PostgreSQL en este caso.

Junto con estos objetivos tecnológicos, también he establecido otro objetivo personal, con un punto de vista más académico, que es el de formarme en nuevas tecnologías como lo son Angular, Spring Boot Framework, Jenkins, Java entre otros.

Con todo ello, este trabajo me permitirá aprender diferentes tecnologías desde diferentes ópticas dado que para la elaboración de este trabajo tendré que ponerme en la piel de un desarrollador de *Backend*, desarrollador de *Frontend* e incluso de técnico de sistemas, que son los encargados de subir y mantener las aplicaciones en un servidor.

1.2. Aportaciones al ámbito empresarial.

Debido a la temática a la que se dedica esta empresa, sus trabajadores están cambiando de proyecto continuamente. Anteriormente, al disponer de una plantilla más reducida, resultaba relativamente fácil controlar y reasignar y adaptar a los diferentes desarrolladores de software que se quedaban sin proyecto con las necesidades de los nuevos proyectos, sin embargo, debido al incremento de plantilla, y al incremento en el número de proyectos, a fecha de hoy se ha convertido en una tarea más compleja de lo habitual, y la expectativa para el futuro indica que esta nueva situación será la tónica habitual.

Por este motivo es necesario el desarrollo de una aplicación que permita de forma rápida y sencilla consultar información sobre cualquier trabajador de la empresa, incluso sobre trabajadores que ya no estén en la empresa.

La aplicación creada en este proyecto permitirá además realizar una búsqueda masiva por skills, buscando cubrir en su totalidad, y de forma efectiva, las necesidades de nuevos proyectos.

1.3. Estructura del documento.

En este apartado se expone un resumen de cada uno de los capítulos que conforman el documento.

- Capítulo 2. Estado del Arte: en este capítulo se comparan las tecnologías que se han utilizado. Para cada una de ellas, si es posible, se plantean posibles alternativas, explicando sus ventajas, desventajas y finalmente justificando porqué se ha escogido una tecnología frente a otras.
- Capítulo 3. Recursos utilizados: en este apartado se describen las herramientas, la documentación y las tecnologías utilizadas.
- Capítulo 4. Desarrollo del proyecto: en este capítulo se detallan las diferentes fases de desarrollo del proyecto, así como el análisis, diseño e implementación del mismo.
- Capítulo 5. Conclusiones y trabajo futuro: En este último capítulo se relatan las conclusiones finales, y se delimitan las futuras ramas de trabajo para la mejora de la aplicación.
- Apéndice A. Manual de usuario: En este apartado se expone un breve manual de uso de la aplicación.

Capítulo 2 - El Estado del Arte

En este capítulo se presentan las tecnologías empleadas para la elaboración de este proyecto.

En primer lugar, se dará una breve definición de un término ampliamente conocido como "la Web". Una vez entendido este concepto, se continuará explicando en qué consiste una consulta web. Finalmente, el capítulo concluye con una sección en la que se exponen las tecnologías que se han usado en el proyecto, junto con una comparación con tecnologías de similares características. También se incluye una breve justificación de por qué se ha escogido una tecnología frente a otras otra.

World Wide Web, también conocido como "la Web", es un sistema de páginas web interconectadas que se emplea en todo el mundo. Nació a finales de los 80, y gracias a ella, con una conexión a Internet y un ordenador o Smartphone, cualquier persona puede acceder a cualquier web que desee desde cualquier parte del mundo.

La web está compuesta de infinidad de documentos que están conectados entre sí a través de enlaces. Habitualmente, cuando un usuario desea acceder a una página web, debe introducir la URL en el navegador, esto quiere decir que se está solicitando la muestra de una página web. Una URL es una dirección o nombre que se le asigna a una página web o recurso que esté almacenado en la Web. Este nombre es único.

Un ejemplo sería www.google.es o www.facebook.com

Una vez introducido en el navegador la URL, el siguiente paso, consiste en verificar qué tipo de petición es la que se está haciendo. Si se trata de acceder a una página web se haría uso de una petición GET que obtiene el sitio. Si se trata de enviar información al sitio web, como, por ejemplo, si alguien se estuviera intentando identificar como usuario de una página web, se trata de una petición POST. Otro tipo de petición podría ser si el usuario se quisiera dar de baja en la misma en ese caso se estaría utilizando una petición DELETE.

Finalmente, existen dos componentes que se encargan de dos partes bien diferenciadas en: Por un lado, el *Backend* se encargará de gestionar la información en base a la petición http recibida; y por otro lado, el *Frontend* muestra la información que le transmitió el *Backend*. Esto es lo que se muestra en el navegador.

2.1. API REST o API RESTFull

Para el desarrollo de este punto nos hemos apoyado en un canal de *Youtube* llamado "HTML Rules", el cual contiene muchísima información sobre tecnologías del sector.

Para comprender lo que es un API RESTfull [\[2\]](#) es conveniente, en primer lugar, entender lo que es un *web service*, y posteriormente, lo que es un API, ya que una vez que se hayan interiorizado estos dos conceptos se podrá comprender lo que es un API RESTfull.

2.1.1. Web Services

[\[2\]](#) Se entiende por servicio Web un marco de comunicación entre dos ordenadores. En dicha comunicación un equipo será el cliente y el otro el servidor. Estos equipos se comunican a través de internet, donde el cliente envía una solicitud, el servidor la recibe, la procesa y devuelve una respuesta al cliente.

Cuando el navegador realiza una petición acerca de una página web recibe en HTML, pero cuando sólo solicita datos, se utiliza JavaScript u otro tipo de código para procesar la respuesta. Los servicios web suelen tener requisitos muy estrictos para el uso, y estos requisitos están documentados para ser usados por los desarrolladores. Las comunicaciones en los Servicios Web modernos se manejan a través de HTTP (Hypertext Transfer Protocol). Sin embargo, el formato del mensaje que se está enviando y recibiendo puede diferir mucho.

Existen dos tecnologías fundamentales para el desarrollo de servicios web: SOAP y REST. En la actualidad se está llegando a la conclusión de que SOAP resulta demasiado complejo, por lo que se está comenzando a utilizar Servicios Web basados en REST para mostrar cantidades de datos masivos.

2.1.2. API

API es el acrónimo de *Application Programming Interface*, y no es más que el software que permite que otros programas se comuniquen con un programa específico. A diferencia que los Web Services, las API no necesitan comunicarse con una red. Podrían usarse entre dos aplicaciones dentro de un mismo ordenador.

2.1.3. SOAP vs REST

En primer lugar, SOAP es el acrónimo de *Simple Object Access Protocol*. Se trata de un lenguaje basado en XML que representa la petición y respuesta de mensajes entre clientes y servidores. SOAP fue desarrollado por Microsoft en la década de los años 90 por un grupo de desarrolladores dirigido por Dave Winer, y desde el año 2000 se gestiona por la World Wide Web Consortium (W3C).

Este lenguaje es la tecnología fundamental que subyace a todo un conjunto de especificaciones conocidas como los estándares WS, y resulta el "pegamento" que hace de enlace entre los juegos de herramientas de servicios web disponibles para muchos sistemas operativos y lenguajes de programación.

SOAP tiene tres objetivos principales en su desarrollo.

- Extensibilidad. A diferencia de otros lenguajes XML, SOAP no tiene como objetivo un tipo específico de proceso de negocio, sino que trata de un lenguaje de programación que está diseñado para manejar los procesos y tipos de datos que se requieran para realizar cualquier trabajo.
- Neutralidad. Aunque los mensajes de SOAP se envían y se reciben comúnmente a través de HTTP, también se pueden utilizar otros protocolos de transporte, tales como por ejemplo SMTP, FTP u otros.
- Independencia. Originalmente el kit de herramientas de SOAP fue construido para lenguajes de programación apoyados por Microsoft, como Visual Basic o C++, sin embargo, pronto se crearon kits de herramientas para otros lenguajes y plataformas.

Actualmente, se encuentran dos versiones de SOAP en uso: SOAP 1.1 y SOAP 1.2.

SOAP 1.1 fue lanzada en el año 2000 y SOAP 1.2 se lanzó en 2007, pero la mayoría de las principales herramientas dan apoyo para ambas. Por ejemplo ASP.net de Microsoft es compatible para servicios web que envían mensajes tanto en la primera versión como en la segunda para así maximizar la interoperabilidad entre los servidores y clientes disponibles.

La segunda versión tuvo muchas mejoras, incluyendo un procesamiento más claro y modelos de extensibilidad. Además, también da una mejor integración con la web, por ese motivo para la mayoría de nuevas aplicaciones acaba predominando el uso de SOAP 1.2.

Por otro lado, cabe añadir que cuando un desarrollador trabaja con SOAP se le libera de toda la responsabilidad de la serialización y deserialización de mensajes XML, ya que todo es manejado por el kit de herramientas SOAP o las bibliotecas que estén usándose dentro de la aplicación.

Esto implica que cualquier desarrollador necesitará tener una biblioteca que sea compatible con su plataforma y con su lenguaje de programación. Necesitaría indicar el servicio que desee usar proporcionando un documento en un formato XML, y luego llamaría al objeto y métodos del servicio remoto como si estuvieras llamando a funciones locales. Por último, los mensajes se envían de ida y vuelta en un formato conocido, y cuando los datos regresen del servicio web, son analizados por la biblioteca del cliente y devueltos al medio local como datos nativos.

Como último punto sobre SOAP antes de hablar de REST, cabría destacar alguna de sus ventajas, tales como estar apoyado ampliamente por muchas plataformas y lenguajes, librándose así al programador de tener que tratar directamente con la creación o lectura de mensajes de servicios web, ya que serían sus mismas herramientas las que traducen los datos hacia nativos y viceversa. De igual manera que cuando se hace con HTTP funciona con la estructura de internet ya existente. Por otro lado, la principal desventaja sería el tamaño: XML es más detallado que otros formatos de mensaje como por ejemplo JSON, por lo que toma más tiempo en procesarse y la traducción de datos nativos a XML y viceversa también se acaba ralentizando.

Tras haber expuesto estos puntos más significativos sobre SOAP, es el turno de REST.

REST significa *Representational State Transfer*. A diferencia de SOAP, que en sentido estricto es un formato de mensaje en un servicio web en particular, REST es un estilo de arquitectura de software que puede ser utilizado con muchos formatos de lenguaje. Los servicios web que siguen el estándar REST muchas veces se conocen como RESTfull. A modo de contextualización histórica, es interesante resaltar que REST fue definido originalmente en los años 2000 por un científico llamado Roy Fielding, que también contribuyó al desarrollo de los HTTP.

Los objetivos principales de REST son:

- Escalabilidad. Un sistema basado en REST debe trabajar bien con muchos clientes y muchas transacciones.
- Generalidad en los interfaces. Deben ser adaptables a cualquier proceso de negocio.
- Despliegue independiente de componentes. Al igual que SOAP, el cliente y el servidor en una conversación basada en REST no necesitan utilizar el mismo sistema operativo o incluso el mismo lenguaje de programación. Además, un servicio basado en REST puede depender de componentes intermedios para reducir la latencia, reforzar la seguridad y encapsular los sistemas generados.

Los servicios RESTfull esperan peticiones en forma de URI (Identificadores Uniformes de Recursos) y estos URI tienen ciertos parámetros particulares. El cliente tiene que formar las solicitudes basadas en las especificaciones del servicio y pasar los parámetros para que se puedan interpretar de manera apropiada. Esto es muy diferente de SOAP, donde las solicitudes siempre se construyen como paquetes XML. Las peticiones REST resultan más pequeñas y concisas, y no contienen toda la información y los metadatos que puede contener una solicitud SOAP.

De nuevo, a diferencia de SOAP, que en su esencia es un formato de mensajería muy específico, los datos devueltos por un servicio RESTfull puede estar en cualquier formato, los más comunes son XML y JSON. Los datos XML, además, podrían estar en cualquier dialecto XML, sin embargo, es más común encontrarlos en formato ATOM, incluso a veces, los podemos encontrar en formato RSS.

Los servicios RESTfull de nueva generación tienen cierta tendencia a devolver datos representados como JSON. La ventaja de JSON es que tiene menos metadatos y menos acotaciones decorativas, de modo que son paquetes mucho más pequeños que sus equivalentes en XML. Además, Los servicios RESTfull también pueden devolver valores que no sean de texto, como archivos binarios, imágenes, documentos PDF, etc.

Para finalizar con este punto, se aprecia que, en conclusión, REST, a diferencia de SOAP, permite muchos más formatos de datos diferentes, ya que SOAP solo permite XML. Además, se puede afirmar que tiene un mejor rendimiento y una mejor escalabilidad. Mientras que en SOAP las lecturas no se pueden almacenar en caché, REST sí lo permitiría. También tiene más eficiencia, ya que REST puede usar formatos más pequeños (tales como HTML, JSON etc.), mientras que SOAP solo usa XML para todos. Además, no requiere de procesamientos externos, por lo que le hace obtener una mayor rapidez.

2.2. Tecnologías Backend.

En primer lugar, sería conveniente saber a qué se hace referencia cuando se habla de este concepto. *Backend* [3] es la capa de acceso a datos de la que dispone un software o aplicación, la cual no es accesible para los usuarios.

Algunos de los lenguajes de programación de *Backend* serían Python, PHP, Ruby, C# y Java. En la figura 1. Se muestran más tecnologías que son consideradas tecnologías *Backend*. Cada una tiene diferentes frameworks que le permiten trabajar con mayor eficacia en función del proyecto que se esté desarrollando.

En los siguientes apartados se realizará una comparativa entre Java y C# ya que son los dos lenguajes de programación orientados a objetos más populares; Spring y Django por ser dos de los frameworks más populares actualmente, a pesar de que el primero trabaja sobre Java, y el segundo sobre Python; Maven y Gradle por ser las herramientas de estandarización de Java más populares; y PostgreSQL y MySQL por ser dos de los sistemas de gestión de base de datos más populares.



Figura 1: Algunas tecnologías de Backend

2.2.1. Java vs C#

Java[4] es un lenguaje de programación de propósito general, es decir, se utiliza para aplicaciones web, empresariales, móviles, etc. Además, es un lenguaje concurrente, lo que quiere decir que puede ejecutar varias tareas a la vez. Se puede decir que Java a fecha de hoy es el lenguaje de programación más popular.

Fue desarrollado inicialmente por un equipo dirigido por James Gosling en Sun Microsystems. En un principio se creó para programar pequeños dispositivos electrónicos. Se pretendía crear un lenguaje similar a C++, pero que consumiera menos recursos y estuviera mejor estructurado, lo que implicaría menores tiempos de desarrollo y un menor número de errores en la programación.

Java es multiplataforma, es decir, que un programa en Java podrá funcionar en cualquier ordenador del mercado gracias a la creación de una Máquina Virtual de Java para cada plataforma. Esto hace de puente para que el sistema operativo y el programa de Java puedan entenderse sin problemas.

Las características de Java son:

- Es orientado a objetos. El paradigma de programación orientada a objetos supuso un gran avance en el desarrollo de aplicaciones, ya que es capaz de acercar la forma de programar a la forma de pensar del ser humano.
- Es relativamente fácil de aprender comparado con otros lenguajes.
- Viene acompañado de una serie de librerías estándar para realizar multitud de operaciones comunes a la hora de programar.
- Más rápido que otros lenguajes. Es más rápido que los lenguajes de programación interpretados como Python, y dentro de los lenguajes de programación compilados no es de los más lentos.
- Es multiplataforma. Esto quiere decir que Java funciona prácticamente en cualquier dispositivo, servidor o sistema operativo, y es una de las principales características por la que se decanta mucha gente.

Es importante resaltar que en este proyecto se ha elegido esta tecnología debido a la familiaridad y control ya adquirido sobre la misma.



Figura 2: Logo de Java

Una vez vistos los puntos fuertes y débiles de Java se puede dar paso a uno de sus principales competidores, como es C#[5], el lenguaje líder en la plataforma Windows. Es la parte principal de todas las herramientas de desarrollo de Microsoft.

C#, al igual que java, es un lenguaje de programación orientado a objetos, pero en este caso fue creado por Microsoft en el año 2001.

Las características de C# son:

- Sintaxis sencilla. Al igual que Java, C# tiene una sintaxis muy sencilla, lo que lo hace un lenguaje realmente fácil de aprender.
- Escritura. Al ser un lenguaje orientado a objetos obliga que todos los métodos y propiedades estén dentro de una clase.
- Orientado a componentes. C# permite definir propiedades sin crear métodos, o usar eventos sin tratar con punteros. Escribir en C# es más simple que con otros lenguajes.
- Sistema de tipos unificado. Todos los tipos de datos sencillos en C# se encuentran en una clase llamada System.Object. La conversión de tipos sólo se permite entre los tipos más cercanos.
- CLR. Ayuda a la integración del sistema operativo o dlls externas.
- Espacio de nombres. C# permite agrupar código mediante namespaces, parecidos a los package de Java.
- Multiplataforma. EL lenguaje C# es de código abierto y se ha utilizado en otros IDEs.
- Tratamiento de errores. Al igual que Java, utiliza las excepciones para controlar posibles errores en el código.
- Multihilo. Posibilidad de dividir tu código en múltiples hilos de ejecución, trabajar en paralelo y sincronizarse al final.



Figura 3: Logo de C#

2.2.2. Spring vs Django.

En primer lugar, Spring[6] es un framework de código abierto para la creación de aplicaciones empresariales de Java. Tiene una estructura modular y una gran flexibilidad para implementar diferentes tipos de arquitecturas según las necesidades de la aplicación.

Un framework, a diferencia de una librería, es un conjunto de artefactos software, conceptos y metodologías, que nos provee de un mecanismo genérico para resolver uno o más problemas. Es extensible mediante el código de usuario y nos facilita las tareas de desarrollo y despliegue.

Spring funciona sobre JVM e inicialmente se diseñó para trabajar con Java, pero, a día de hoy, se ha visto que puede trabajar con Groovy, que es un lenguaje de programación implementado sobre Java y con características similares a Perl, Python SmallTalk y Ruby. También tiene soporte con Kotlin, que es otro tipo de lenguaje de tipado estático que corre sobre JVM y que puede ser compilado a Javascript.

En lo que respecta a su estructura, Spring la tiene modular. Esto quiere decir que la conforma una familia de proyectos que abarca múltiples ámbitos: desarrollo de aplicaciones web, aplicaciones web reactivas, seguridad, servicios web, micro servicios, Android etc.

Una de sus características más versátiles es que utiliza *Inversion of Control* invirtiendo el control del programa y externalizando la construcción y el manejo de objetos. De esta manera Java solventa el problema del manejo de memoria, ya que, al invertir el control del programa, éste no estará continuamente consultando si necesita realizar cualquier operación, si no que él mismo lo avisará cuando sea necesario.



Figura 4: Logo de Spring

Por otro lado, Django[7] es un framework web gratuito de código abierto y escrito en Python que respeta el patrón de diseño conocido como Modelo-Vista-Controlador. Fue liberada al público en julio de 2005, y en un principio, se desarrolló para gestionar varias páginas orientadas a noticias.

La meta fundamental de Django es facilitar la creación de sitios web complejos, poniendo énfasis en el re-uso, la conectividad y extensibilidad de componentes y en el desarrollo rápido.

Python es usado en todas las partes del framework, incluso en configuraciones, archivos y en los modelos de datos.

Las características de Django son:

- Presentación: aquí se maneja la interacción entre el usuario y el computador. El sistema de configuración de URL también es parte de la capa de presentación.
- Control: en esta capa reside el programa o la lógica de aplicación en sí. En Django son representados por las *views* y *manipulators*. La capa de presentación depende de ésta, y a su vez, ésta depende de la capa de dominio.
- Mediator: es el encargado de manejar la interacción entre el subsistema *entity* y *foundation*. Aquí se realiza el mapeo objeto-relacional a cargo del motor de Django.
- Entity: el subsistema entity maneja los objetos de negocio. El mapeo objeto-relacional de Django permite escribir objetos de tipo entity de una forma fácil y estándar.
- Foundation: la principal tarea del subsistema foundation es la de manejar a bajo nivel el trabajo con la base de datos. Se provee soporte a nivel de foundation para varias bases de datos y otras están en etapa de prueba.
- Django también cuenta con: servidor web incluido para entorno de desarrollo, potente ORM, sistema de plantillas, pretty urls, base de datos embebidas, etc.



Figura 5: Logo de Django

2.2.3. Maven vs Gradle.

En primer lugar, Apache Maven[8] es una herramienta que estandariza la configuración de un proyecto. Basándose en el concepto de modelo de objeto de proyecto (POM), Maven puede administrar la construcción, los informes y la documentación de un proyecto a partir de una información central.

Además, es la base de los compiladores actuales (Eclipse, NetBeans, IntelliJ), ofreciendo soporte para cubrir el sistema de gestión de dependencias, un mecanismo de distribución de librerías y mecanismos para ser extensible, además de ser una herramienta multiplataforma.

Principales comandos de Maven

Para ejecutar estos comandos es necesario colocarse en el directorio donde se encuentra nuestro fichero pom.xml.

- **mvn clean:** borra todos los .class, .jar y .war generados.
- **mvn compile:** ejecuta tarea de compilación de proyecto.
- **mvn package:** genera el .war o .jar del proyecto.
- **mvn install:** lleva el .jar o .war generado al repositorio local quedando accesible para otros proyectos de Maven dentro del ordenador.
- **mvn deploy:** lleva el .jar o .war generado al servidor de jar y/o war de forma que el proyecto sea accesible por otros ordenadores.

Características de Maven:

- Es un sistema de gestión de dependencias.
- Es extensible gracias a la creación de plugins.
- Es multiplataforma, ya que su funcionamiento es posible tanto en entornos de Linux, como Windows, gracias a ser una aplicación Java.
- Es de código abierto, con lo que se podría modificar y customizar por cualquier desarrollador.
- Fomenta la reutilización de código y librerías: ofrece repositorios oficiales y públicos de software libre a toda la comunidad de desarrolladores, y a través de estos repositorios los desarrolladores de un mismo proyecto pueden compartir sus librerías.

- Es compatible con múltiples IDEs de programación.



Figura 6: Logo de Maven

Por otro lado, Gradle[9] es una herramienta de código abierto, y se centra en dos pilares: flexibilidad y rendimiento.

Este se puede programar tanto en Groovy como en Kotlin y su principal característica es su gestión de dependencias, ya que puede presumir de tener una de las mejores. Respecto a su flexibilidad, permite programar varios lenguajes, ya sea Java, Swift, etc. Posee un lenguaje DSL (Domain Specific Language) que simplifica mucho la programación, y además cuenta con integración a terceros, ya que se puede coger un proyecto Gradle e integrarlo con otro fácilmente.

Características de Gradle:

- Depuración colectiva: es capaz de compartir los resultados de la compilación en diferentes equipos para poder solventar posibles errores de forma colectiva.
- Embalaje y distribución de JAR, WAR y EAR: contiene herramientas para empaquetar el código de forma rápida y sencilla en formatos de archivo comunes.
- Soporte de MS Visual C++ y GoogleTest.
- Publicar en repositorios Maven: Gradle permite una publicación sencilla de sus Artifacts en repositorios Ivy o Maven Central.
- Lee formato POM: es compatible con el formato de metadatos POM, lo que permite utilizar dependencias de repositorios compatibles con Maven.



Figura 7: Logo de Gradle

2.2.4. PostgreSQL vs MySQL

Antes de explicar qué es PostgreSQL [10] y MySQL [11], es necesario tener claro qué es SQL. Se trata de un lenguaje de consultas estructurado, declarativo y con acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.

Las bases de datos relaciones son la forma más común de almacenar datos en las aplicaciones web. Permiten crear relaciones entre diferentes tablas y diferentes elementos mediante una sintaxis sencilla y asequible que permite recuperar estos datos y poder devolverlos a la web.

PostgreSQL es un sistema de gestión de bases de datos relacional, orientado a objetos, dado que todos los elementos de nuestra base de datos van a poder tratarse como objetos. Es multiplataforma, extensible, se le pueden añadir funcionalidades, es escalable y funciona bajo licencia libre.

Además, se usa en diversos ámbitos, por ejemplo, almacenamiento de datos, procesamiento de datos, sistemas de información geográfica, bases de datos para servicios web, etc.

Características de PostgreSQL:

- Lenguaje SQL: es muy próximo al estándar ISO/IEC, lo que significa que resultará relativamente sencillo portar consultas y scripts de otros sistemas de bases de datos.
- Cumple con ACID: provee atomicidad, consistencia, aislamiento y durabilidad para sus operaciones.
- Permite crear: esquemas, tablas heredadas o triggers orientados a eventos que no poseen otros motores.
- Permite definir procedimientos: no solo en PostgreSQL, sino también en otros muchos lenguajes como Pearl, TCL o Python.
- Soporta tipos de datos definidos por el usuario: en el caso de que el dato no esté importado de serie.
- Es posible extender la funcionalidad: con extensiones provistas por la propia PostgreSQL, por terceros, o incluso programándolas por cuenta ajena.
- Soporte nativo de replicación maestro/esclavo: también es posible añadir otros tipos a través de productos de terceros, libres o de pagos.

- Provee una excelente escalabilidad vertical.

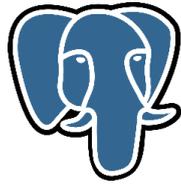


Figura 8: Logo de PostgreSQL

Por otro lado, MySQL es una distribución de SQL gratuita y se ha convertido en la más extendida y popular de toda la web. Sin embargo, es algo más que una base de datos. MySQL es un sistema gestor de base de datos (SGBD) multiusuario, multiplataforma y además de código abierto.

Características de MySQL:

- Cuenta con la capacidad de realizar tareas multiprocesador, debido a que posee la opción de trabajo multihilo.
- Puede ingresar una enorme cantidad de datos por columna de trabajo.
- Cuenta con API's disponibles para los principales lenguajes de programación que existen.
- Cuenta con una portabilidad bastante sobresaliente.
- Capacidad de soportar hasta 32 índices de tablas diferentes.
- Nivel de seguridad que permite gestionar varios usuarios con login y contraseñas individuales.



Figura 9: Logo de MySQL

2.3. Tecnologías Frontend.

Como se ha hecho anteriormente con el *Backend*, ahora hay que entender lo que es Frontend[3]; y es que, si *Backend* se refería a la capa de acceso a datos de un software, el *Frontend* será la capa a la que pueden acceder los usuarios directamente. Lo componen todas las tecnologías de diseño y desarrollo web.

Como puede apreciarse en la figura 10, HTML, CSS y JavaScript son los lenguajes principales de *Frontend*, de los que como con los *Backend*, se desprenden una cantidad de frameworks y librerías que expanden sus capacidades para crear cualquier tipo de interfaces de usuarios.

En los siguientes apartados se hará una comparativa entre Angular y React, ya que son los dos framework de frontend más populares a fecha de hoy.

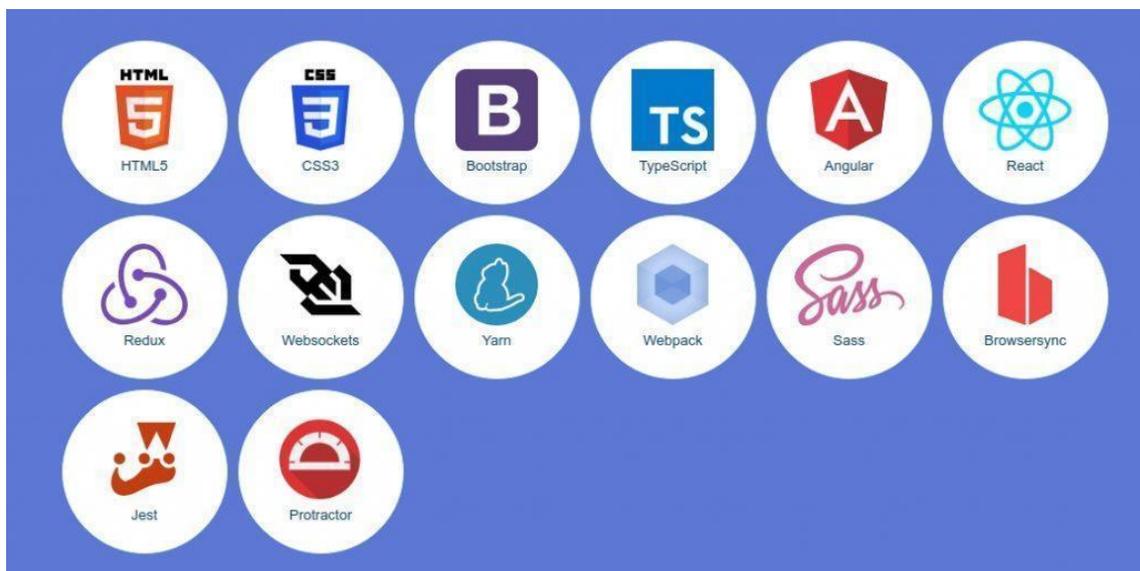


Figura 10: Algunas tecnologías Frontend

2.3.1. Angular vs React.

En primer lugar, Angular[12] es un framework para aplicaciones web desarrollado en TypeScript y es de código abierto mantenido por Google.

Se utiliza para crear y mantener aplicaciones web de una sola página y su objetivo es aumentar las aplicaciones en el navegador, lográndolo al trabajar con la arquitectura Modelo Vista Controlador (MVC). Angular es la evolución de AngularJS, a pesar de que no sean compatibles.

A continuación, se citarán las principales características de Angular, destacando entre ellas su división y generación del código gracias al uso de componentes y enrutadores, su universalidad y su manejo de plantillas, y además cuenta con una herramienta de línea de comandos llamada Angular CLI, que permite empezar a desarrollar aplicaciones de forma rápida.

Características de Angular:

- Permite usar TypeScript, pudiendo potenciar sus características y siendo la opción más recomendada, especialmente cuando se piensa en escalabilidad.
- Angular es muy útil para construir páginas web SAP (Single Page Application).
- Es modular, ya que se basa en módulos y en un core que permiten acceder a más características cuando es necesario.
- Permite crear componentes.
- Tiene fácil integración con otras tecnologías, ya que es posible generar un proyecto desde el CLI con Node.js y, por ejemplo, trabajar las vistas con React.



Figura 11: Logo de Angular

Por otro lado, React[13] es una librería de código abierto desarrollada por Facebook para la creación de interfaces de usuario con el objetivo de animar al desarrollo de aplicaciones de una sola página.

React intenta ayudar a los desarrolladores a construir aplicaciones cuyos datos están en continuo cambio. Pretende ser sencillo, declarativo y fácil de combinar. Sólo maneja la interfaz de usuario en una aplicación y está construida para utilizar el patrón de diseño modelo-vista-controlador (MVC). Además, puede ser utilizada conjuntamente con otras bibliotecas de Javascript (o más grandes como AngularJS), y puede ser utilizada con las extensiones de React-based, las cuales se encargan de las partes no gráficas de una aplicación web.

Características de React:

- Renderiza todo con cada cambio.
- Virtual DOM: modificar el DOM es una operación costosa que React hace por nosotros de una manera sencilla, tratándolo como si de una GPU se tratara.
- Eventos sintéticos: React crea un único manejador de eventos nativo en el nivel superior de la estructura de cada componente. Este sistema de eventos está normalizado para que sea funcional en todos los navegadores e incluso es capaz de diferenciar entre eventos *desktop* y móviles.

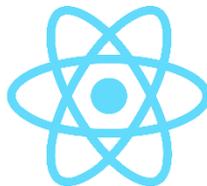


Figura 12: Logo de React

Capítulo 3 – Recursos utilizados

En este capítulo se van a introducir los recursos que se han utilizado para la elaboración de este proyecto, desarrollando los softwares y las librerías sobre las que está apoyado.

3.1. Software.

3.1.1. Eclipse

En lo que respecta al software, mencionar en primer lugar a Eclipse[\[14\]](#), famoso por su entorno de desarrollo integrado (IDE) de Java y por contar con un número de IDEs bastante considerable donde se incluyen IDE de C/C++, de JavaScript/TypeScript y de PHP, entre otros.

Como comentario personal, decir que se trata de un software bastante sencillo de usar, y fue el primer IDE que utilicé en mi ordenador, es por eso que he decidido elegir éste frente a otros IDEs que permiten programación en Java.



Figura 13: Logo de Eclipse

3.1.2. Visual Studio Code

Visual Studio Code[\[15\]](#) es un editor de código fuente disponible para MacOS, Linux y Windows. Es gratuito y de código abierto a pesar de tener un soporte predeterminado para JavaScript, TypeScript y Node.js. Cuenta con una gran variedad de extensiones para poder soportar una gran cantidad de lenguajes.

Además, tiene control integrado de Git, lo que permitirá poder trabajar con él en el proyecto.



Figura 14: Logo de Visual Studio Code

3.1.3. Node.js

Node.js[16] es un entorno multiplataforma y de código abierto, basado en JavaScript. Su principal objetivo es la creación de programas de red altamente escalables como servidores web.

Además, posee un gestor de paquetes llamado NPM que facilita la instalación de librerías, proyectos, etc.



Figura 15: Logo de NodeJS

3.1.4. pgAdmin

pgAdmin[17] es la plataforma más popular tanto para la administración y gestión como para el desarrollo de bases de datos en PostgreSQL. Es de código abierto, válido en diversos sistemas como lo son Windows, Linux, Unix etc. y cuenta con diferentes ediciones, añadiendo nuevas tecnologías como Flask para el Backend, bootstrap o Javascript/Jquery/Backbone para el FrontEnd, entre otros.

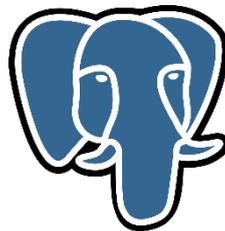


Figura 16: Logo de pgAdmin

3.2. Librerías y Plugins

Este proyecto está apoyado en diversas librerías y plugins, siendo los siguientes:

- Spring Tools 4. Plugin de Eclipse: permite integrar proyectos con el framework de Spring.
- EGit – Git Integration For Eclipse. Plugin que permite la integración de los proyectos con Git, para agilizar la gestión.
- Angular Material[18]. Librería que contiene componentes de diseño de materiales para Angular.
- Ng2SmartTableModule. Librería de tablas de datos inteligentes con funciones de clasificación, filtrado, paginación, etc.
- NgMultiSelectDropDownModule. Librería de componente desplegable de selección múltiple para aplicaciones web.

Capítulo 4 - Desarrollo del proyecto

4.1. Análisis

4.1.1. Requisitos del sistema

Debido a que se trata de una aplicación característica API Rest, el único requisito necesario para su uso es disponer de cualquier dispositivo (móvil, ordenador, tablet...) con acceso a Internet para poder acceder a ella, y así conectarse a una VPN de Capgemini.

4.1.2. Requisitos del software

Este proyecto va dirigido al equipo de Recursos Humanos y Teams Manager de la sede de Capgemini en Murcia, con el objetivo de facilitar diversas tareas como la asignación y reasignación de proyectos o control de desarrolladores, entre otros, para así poder tener un control más exhaustivo.

Además, en un futuro el proyecto podría escalarse para que pueda ser utilizado por más sedes o incluso a nivel nacional.

Casos de Uso

El proyecto debe constar de una página web donde se encuentre toda la funcionalidad, y dependiendo del rango de la persona que utilice la aplicación se tendrá permisos de administrador o permisos de usuario, donde se variará el nivel de dicha funcionalidad.

En las figuras 17 y 18 mostradas más abajo, se exponen dos diagramas de casos de uso donde se refleja la funcionalidad desarrollada por cada uno de estos dos roles (User y Admin).

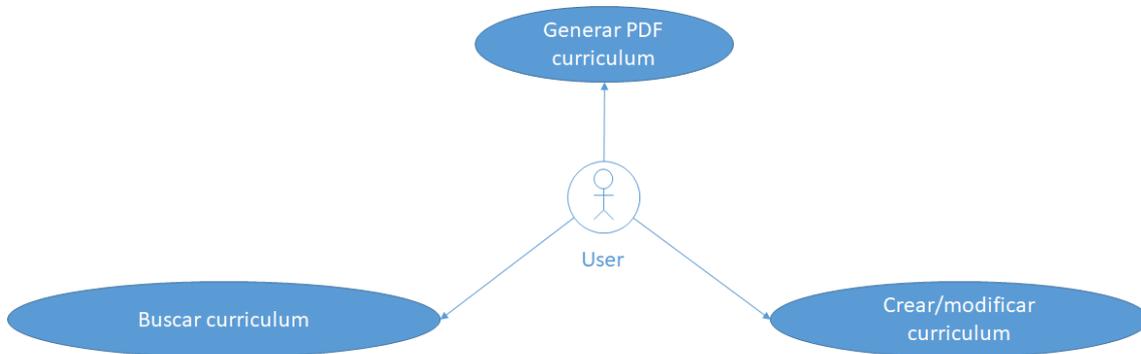


Figura 17: Ilustración de las funcionalidades del servicio que tiene un usuario con rol "user"

Como se puede apreciar, se han desarrollado las funcionalidades básicas de la aplicación como buscar un currículum, crear/modificar un currículum y generar un PDF del currículum para los usuarios con rol "User". Figura 17.

Para los usuarios con rol "Admin" se han ampliado las funcionalidades. Estos usuarios además de crear/modificar, consultar y generar el PDF de un currículum, también podrán crear/modificar/eliminar usuarios, crear/modificar/eliminar skills, eliminar currículums y crear/eliminar/modificar datos comunes como pueden ser Universidades, Grados y Master Universitarios, podrán además modificar que Grados y Máster están asociados a las universidades. También podrán crear/eliminar/modificar Países y Ciudades.

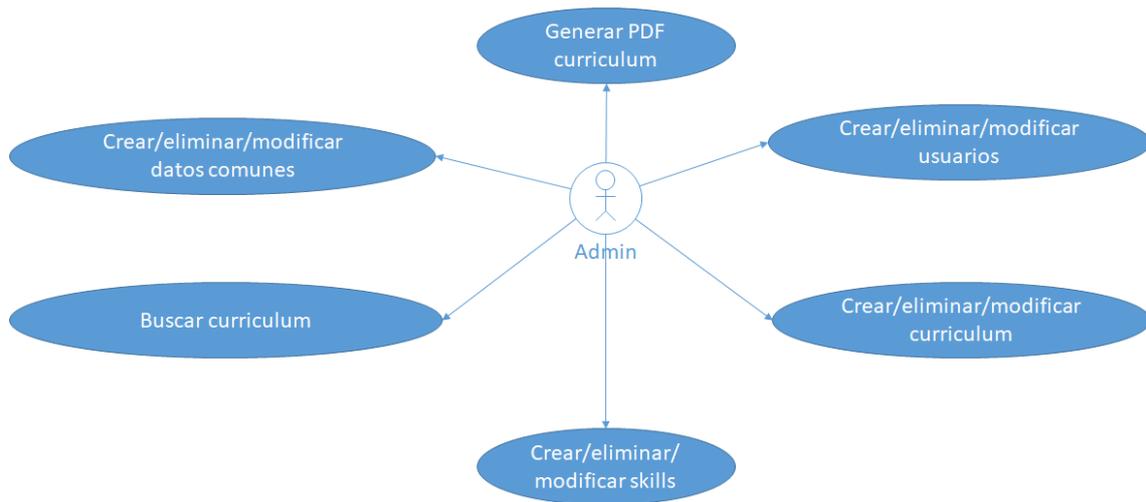


Figura 18: Ilustración de las funcionalidades del servicio que tiene un usuario con rol "admin"

4.2. Diseño

En este apartado se tratan los conceptos relacionados con el diseño de la aplicación, tales como su Arquitectura, estructura de Base de Datos, las interfaces gráficas y también se mostrará el diagrama de clases que sigue la Base de Datos.

4.2.1. Arquitectura

En la figura 19 se muestra la arquitectura del sistema donde se aprecia el flujo de los datos de origen hasta la página web.

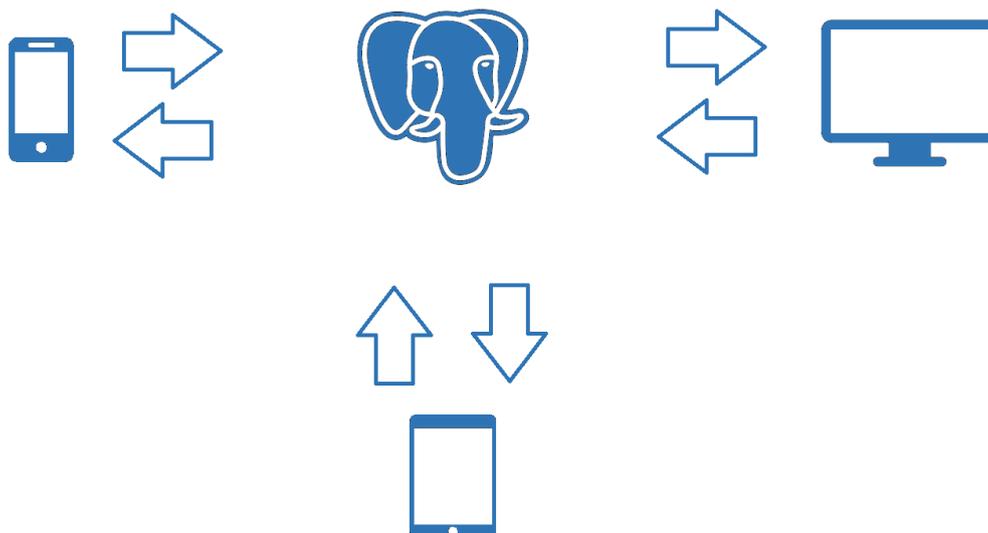


Figura 19: Ilustración del flujo de información a través de los dispositivos.

Se puede ver como el flujo de datos es bidireccional entre los distintos dispositivos, móviles y ordenador. El flujo es bidireccional porque desde cualquier dispositivo podemos registrar un currículum, guardándolo en la Base de Datos de PostgreSQL, y desde cualquier dispositivo podemos hacer una búsqueda de currículum y consultar los datos de cualquier currículum a la Base de Datos de PostgreSQL, y ésta enviará la información al dispositivo.

4.2.2. Base de Datos

Como se puede apreciar en la figura 20, actualmente la base de datos contiene una tabla para cada aspecto, pero su estructura permite ampliarla fácilmente con cualquier otro aspecto nuevo a considerar. Debido a la gran variedad de relaciones OneToOne, OneToMany, ManyToOne y ManyToMany se decidió emplear un modelo de base de datos relacional.

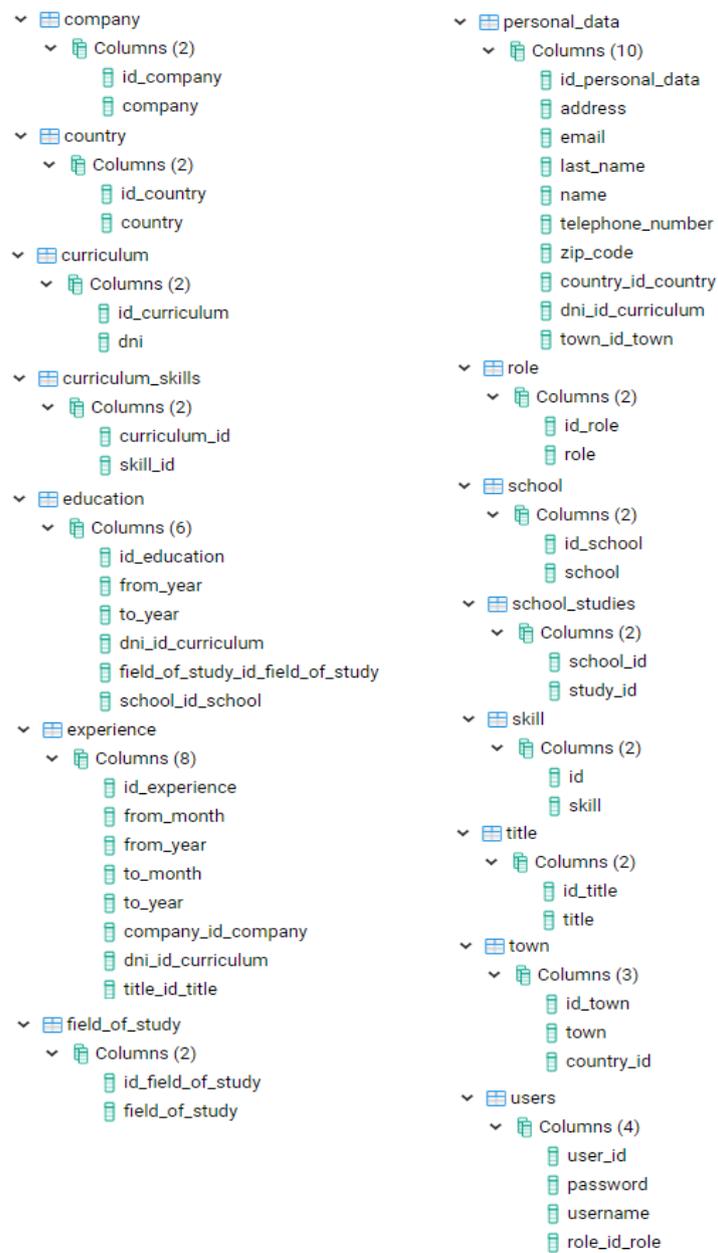


Figura 20: Estructura de la base de datos

4.2.3. Diseño de las interfaces de usuario

En este apartado se muestran algunos diseños de las interfaces gráficas según el rol del usuario. Se comienza con el rol “user” que es el que tiene menos privilegios.

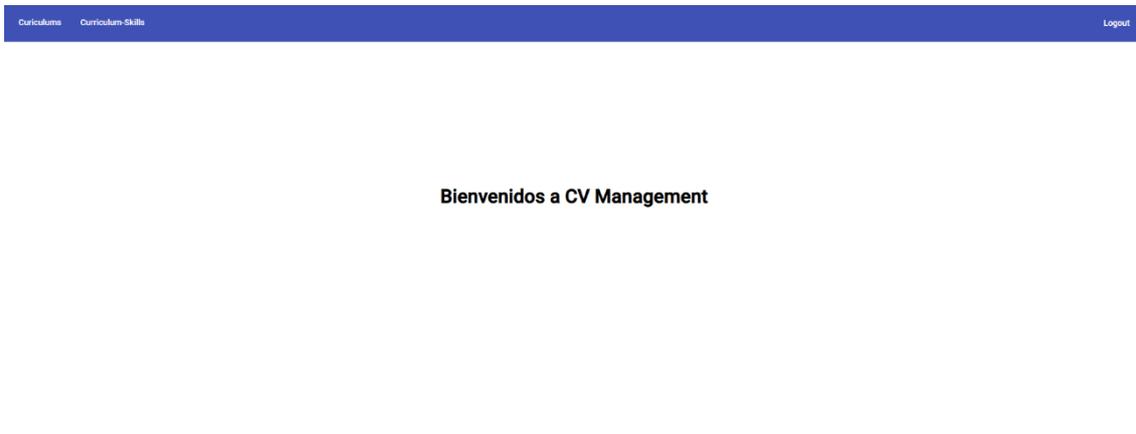


Figura 21: Vista principal de la aplicación del perfil “user”

Como se puede observar, el diseño de la página es bastante sencillo y agradable a la vista. Cuenta con un menú con tres botones: “Curriculums” y “Curriculums-Skills” en la barra superior derecha, y “Logout” en la barra superior izquierda. Si se selecciona el botón “Curriculum” se abriría una ventana con una lista ordenada de todos los curriculums que hay registrados en la Base de Datos. Al pinchar en uno de ellos nos llevaría a la página resumen de su currículum donde aparecerán todos los datos.

Si, por el contrario, se pincha el botón “Curriculums-Skills” se accedería al buscador, donde se pueden filtrar y hacer búsquedas por Skills.

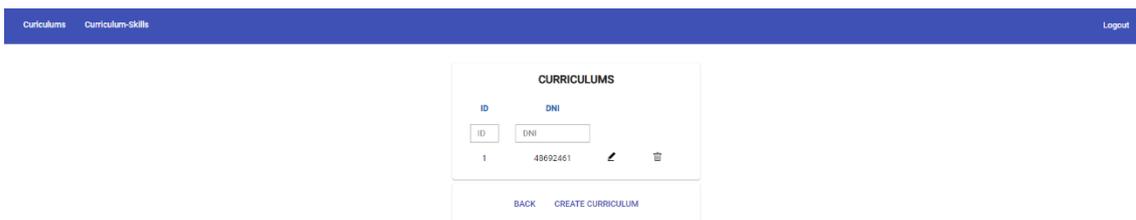


Figura 22: Pestaña “Curriculum”

SKILLS

DNI	Nombre	Apellidos	Skill		
48692461	Jorge	Zapata Alburquerque	Java		
48692461	Jorge	Zapata Alburquerque	Angular7		
48692461	Jorge	Zapata Alburquerque	SOAP		
48692461	Jorge	Zapata Alburquerque	Web Services		
48692461	Jorge	Zapata Alburquerque	Trabajo en equipo		
48692461	Jorge	Zapata Alburquerque	Microsoft Office		
48692461	Jorge	Zapata Alburquerque	Microsoft Word		
48692461	Jorge	Zapata Alburquerque	Gestión de Proyectos		
48692461	Jorge	Zapata Alburquerque	Investigación y Desarrollo		
48692461	Jorge	Zapata Alburquerque	Redes Sociales		

←1234→

BACK

Figura 23: Pestaña “Curriculum-Skill”

A continuación, se muestra el perfil de “admin”, el cual tiene todos los privilegios, y más funcionalidad que el perfil “user”.



Bienvenidos a CV Management

Figura 24: Vista principal de la aplicación del perfil “admin”

Se aprecia que el perfil de usuario “admin” tiene más pestañas que el perfil “user”. La única funcionalidad que tienen estas pestañas adicionales (Las pestañas “Educations”, “School”, “Personal Data”, etc.) es las de administrar, borrar, añadir o editar estos campos, que luego serán utilizados a la hora de registrar un curriculum.

SCHOOLS

ID	School		
<input type="text" value="ID"/>	<input type="text" value="School"/>		
1	CUNEF Universidad		
2	ESIC Universidad		
3	Saint Louis University Madrid Campus		
4	Universidad a Distancia de Madrid		
5	Universidad Abad Oliva CEU		
6	Universitat Oberta de Catalunya		
7	Universidad Alfonso X el Sabio		
8	Universidad Antonio de Nebrija		
9	Universidad Autónoma de Barcelona		
10	Universidad Autónoma de Madrid		

«<-1234->»

[BACK](#) [ADD SCHOOL](#) [RELATE STUDIES](#)

Figura 25: Pestaña "School"

Curriculum Educations Personal Datas Experiences School Studies Skills Companies Titles School-Studies Curriculum-Skills

EXPERIENCES

ID	Company	Title	From Month	From Year	To Month	To Year		
<input type="text" value="ID"/>	<input type="text" value="Company"/>	<input type="text" value="Title"/>	<input type="text" value="From Month"/>	<input type="text" value="From Year"/>	<input type="text" value="To Month"/>	<input type="text" value="To Year"/>		
1	Capgemini	Estudiante en Prácticas	Febrero	2019	Junio	2019		
2	Capgemini	Junior Software Developer	Julio	2019	Abril	2020		

[BACK](#) [CREATE EXPERIENCE](#)

Figura 26: Pestaña "Experiences"

4.2.4. Diagrama de clases

La figura 27 muestra el diagrama de clases que contiene la Base de Datos.

Cabe añadir que el desarrollo de estas clases fue el inicio de este proyecto y que con el propio transcurso y desarrollo del mismo ha ido evolucionando (se han añadido relaciones, cambiado otras, suprimido y añadido tablas etc.). La figura representa el estado actual de dicho diagrama.

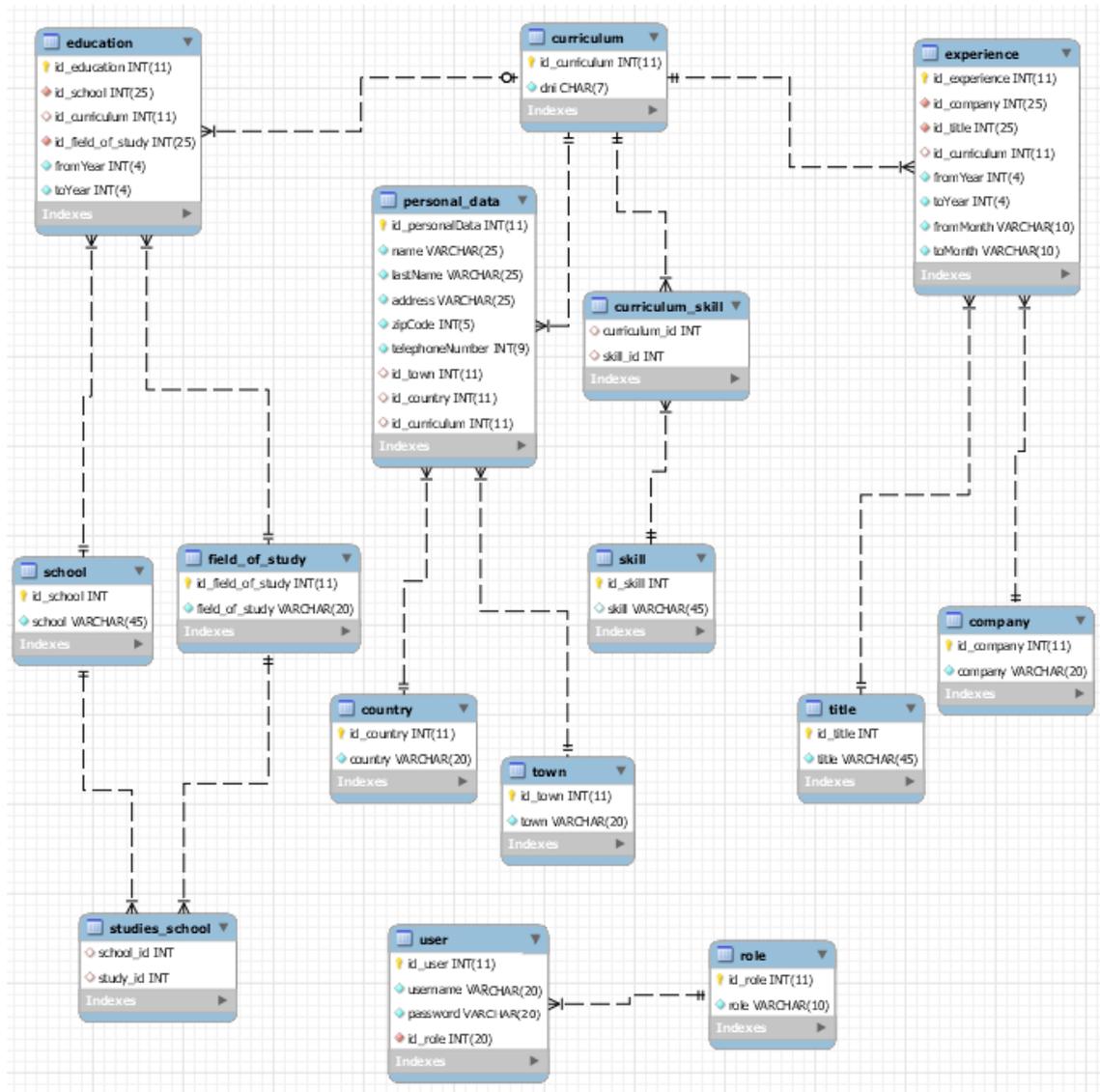


Figura 27: Modelo Entidad - Relación de la aplicación.

4.3. Metodología de trabajo

En este apartado se va a describir la metodología de trabajo, en la que se ha tenido en cuenta el punto de vista de control de tareas, el punto de vista de control de versiones del código y el punto de vista de la implementación en sí.

4.3.1. Kanban

Para llevar un mayor control de las necesidades del proyecto y de las tareas a realizar, durante el desarrollo se ha utilizado la metodología de trabajo Kanban[19]. Se trata de una metodología de trabajo ágil, cuyo propósito se basa en gestionar de manera sencilla cómo se van completando las tareas.

Kanban es una palabra que proviene del japonés, donde “Kan” significa visual y “Ban” tarjetas. De esta manera, la traducción de Kanban sería “tarjetas visuales”. Así pues, como su nombre indica, pretende trabajar con tarjetas, donde se establezcan los objetivos y se haga un seguimiento de ellos.

El programa con el que se ha llevado a cabo esta metodología se llama Trello[20], la cual permite crear Tableros donde cada uno puede ser un proyecto diferente.

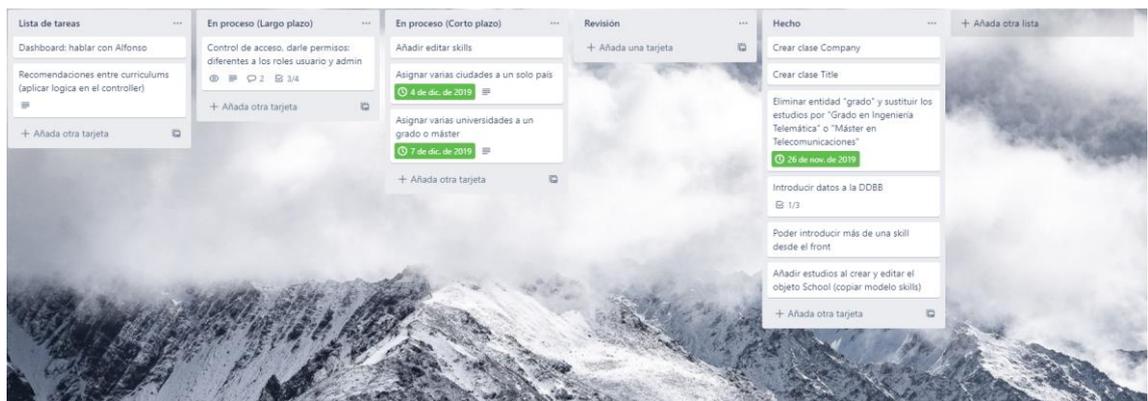


Figura 28: Tablero de Trello de la aplicación

4.3.2. Git

Git[21] es un software de control de versiones que nació en 2005.

Un sistema de control de versiones es un sistema que permite controlar los avances y modificaciones de un proyecto. Permite regresar a través de los cambios hechos en el mismo y recuperar algo del pasado por si hubiera surgido algún problema o simplemente se quisiera regresar sin más. También permite crear “ramas” paralelas al proyecto, donde poder desarrollar códigos sin que afecte a las demás y al final, cuando estén listas, fusionarlas con la rama principal. Además, Git permite trabajar en equipo de forma remota.

4.4. Implementación

En este apartado se van a explicar los pasos realizados para la implementación de este servicio.

Se empezará por el Servidor, explicando y detallando las tecnologías utilizadas y su uso para desarrollar la parte del *Backend*. Posteriormente se explicará el desarrollo de la parte del *Frontend* para la vista visible del cliente. Y por último se hará una pequeña introducción de dos herramientas también utilizadas: una para llevar un control de versiones de la aplicación y así facilitar el trabajo en distintos ordenadores, y la otra para desplegar la aplicación en un servidor de internet.

4.4.1. Servidor

Para programar la parte del servidor de la aplicación o *backend* se ha utilizado principalmente el IDE Eclipse. A este IDE se le ha instalado previamente un plugin para incorporar el Framework de Spring “Spring Tools 4 (Spring Tools Suite 4)”. Para hacerlo: Menú de Eclipse -> Help -> Eclipse Marketplace y buscar sts tools

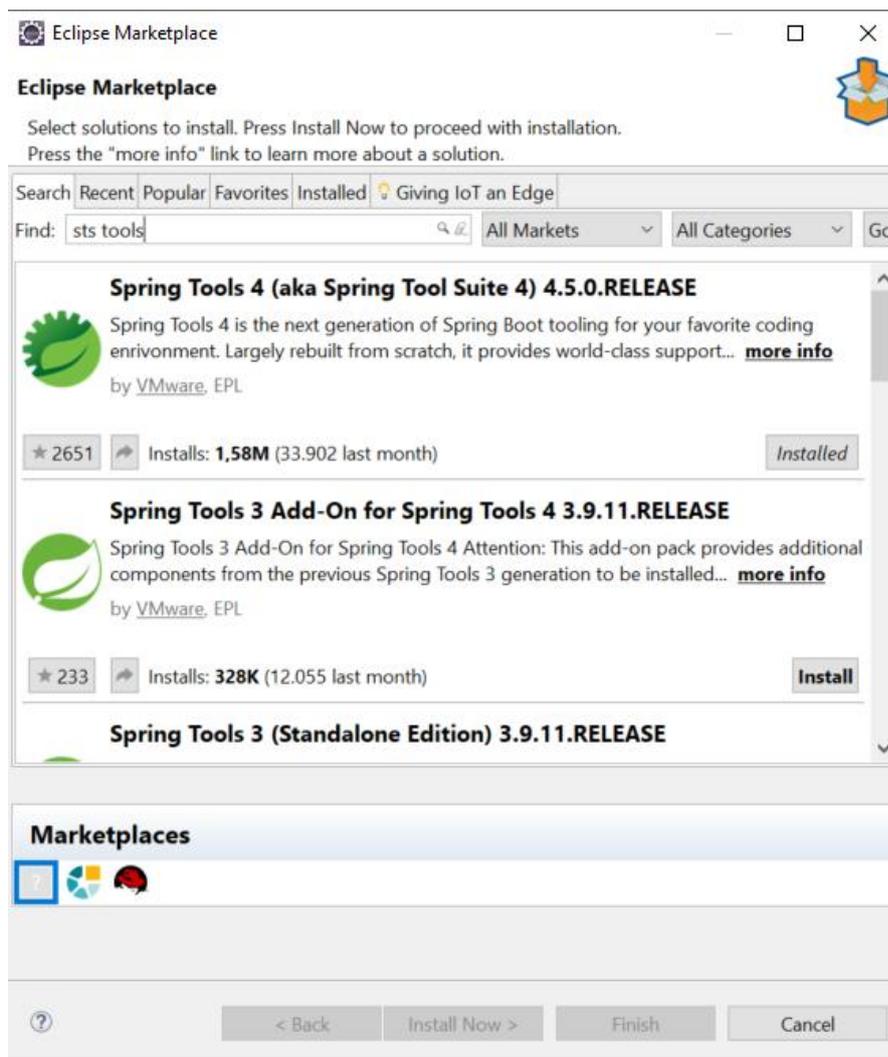


Figura 29: Eclipse Marketplace

Como puede observarse en la figura 30, el plugin que habría que instalar es “*Spring Tools 4 (aka Spring Tool Suite 4) 4.5.0.RELEASE*”. Esperar que se instale y una vez instalado se podrá crear un proyecto Spring.

Para crearlo basta con ir al menú de Eclipse -> File -> New -> Other y buscar “Spring Starter Project”, como muestra la figura 4.14.

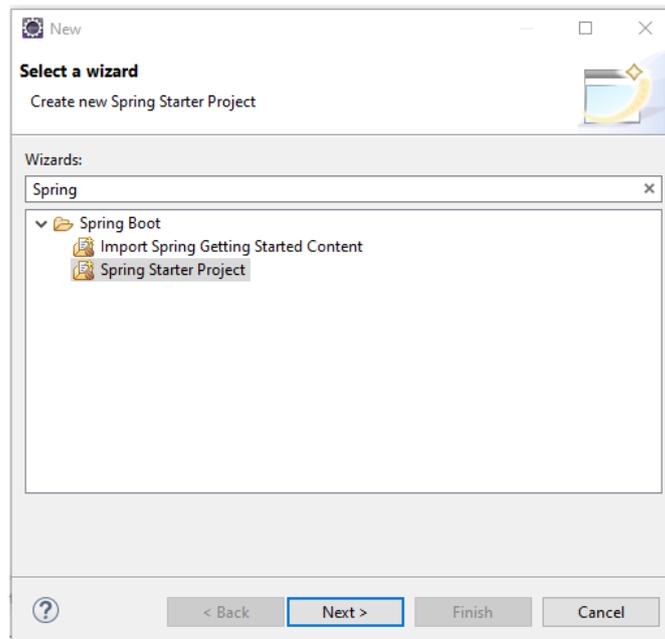


Figura 30: Inicio de un proyecto Spring

Una vez pulsado sobre Spring Starter Project se abrirá una ventana para la configuración del proyecto. Es importante que se seleccione Type Maven, Java Version (la última que se tenga). En este caso Java 8, el Group y el Artifact.

New Spring Starter Project

Service URL:

Name:

Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

Description:

Package:

Working sets

Add project to working sets

Working sets:

Figura 31: Configuración de un proyecto Spring

Si todo ha ido correctamente, se habrá creado un proyecto con un archivo pom.xml como el de la figura 32.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6   <parent>
7     <groupId>org.springframework.boot</groupId>
8     <artifactId>spring-boot-starter-parent</artifactId>
9     <version>2.3.1.RELEASE</version>
10    <relativePath /> <!-- lookup parent from repository -->
11  </parent>
12  <groupId>upct.etsit</groupId>
13  <artifactId>mgcv-1</artifactId>
14  <version>0.0.1-SNAPSHOT</version>
15  <name>mgcv-1</name>
16  <description>Demo project for Spring Boot</description>
17
18  <properties>
19    <java.version>1.8</java.version>
20  </properties>
21
22  <dependencies>
23    <dependency>
24      <groupId>org.springframework.boot</groupId>
25      <artifactId>spring-boot-starter</artifactId>
26    </dependency>
27
28    <dependency>
29      <groupId>org.springframework.boot</groupId>
30      <artifactId>spring-boot-starter-test</artifactId>
31      <scope>test</scope>
32      <exclusions>
33        <exclusion>
34          <groupId>org.junit.vintage</groupId>
35          <artifactId>junit-vintage-engine</artifactId>
36        </exclusion>
37      </exclusions>
38    </dependency>
39  </dependencies>
40
41  <build>
42    <plugins>
43      <plugin>
44        <groupId>org.springframework.boot</groupId>
45        <artifactId>spring-boot-maven-plugin</artifactId>
46      </plugin>
47    </plugins>
48  </build>
49 </project>

```

Figura 32: Archivo pom.xml

Como se observa, el archivo pom.xml sigue una estructura en lenguaje XML. Después de la etiqueta Project se encuentra la etiqueta <modelVersion>. Esto está establecido a 4.0.0 por defecto en Maven 2 y 3. Si en un futuro Maven 4 estableciera su modelVersion en 4.1, los pom que estén bajo ese modelVersion no serían compatibles con Maven 2 y 3.

La siguiente etiqueta que se encuentra es la etiqueta <parent>. Esta etiqueta indica el proyecto padre, del que se desprende nuestro proyecto, en este caso el proyecto padre es el proyecto de Spring que permite arrancar la aplicación.

Luego se encuentran los datos generales del proyecto, el groupId, el artifactId, versión, nombre y descripción del proyecto. Estos datos son los que se han rellenado en el cuadro de la figura 31.

La etiqueta <properties> establece propiedades dentro del proyecto. En este caso sólo tiene la propiedad de la versión de java.

A continuación, está la etiqueta <dependencies>. Aquí dentro irán todas las dependencias de las librerías que se quieran integrar en el proyecto. Para agregar una librería basta con abrir una etiqueta <dependency> y dentro establecer el <groupId>, <artifactId> y <version> de la dependencia que se quiera añadir. Si se desconocen estos datos, se puede visitar <https://mvnrepository.com/> y buscar ahí dicha dependencia.

Como se puede apreciar en la figura 32, al inicializar el proyecto como proyecto Spring, se han añadido ya 2 dependencias: spring-boot-starter y spring-boot-starter-test.

Por último, se encuentra la etiqueta <build> y dentro de ésta, la etiqueta <plugins>. Aquí irán los plugin que se quieran añadir a nuestro proyecto. En este caso ya está inicializado con un plugin “spring-boot-maven-plugin”.

Una vez explicada la estructura del archivo POM.xml, se va a explicar la estructura del *backend* en este proyecto. La estructura sería la que se muestra en la Figura33.

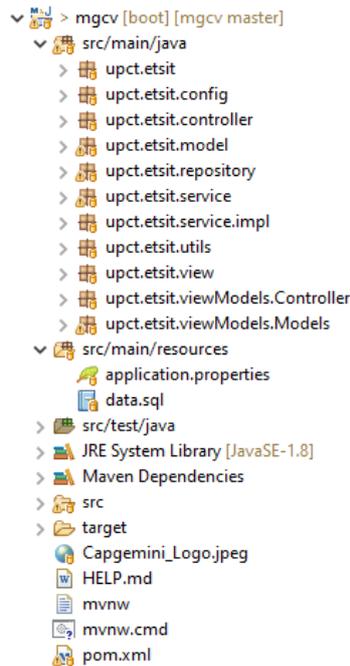


Figura 33: Arquitectura del Backend

Como se puede apreciar, el proyecto se divide en 11 paquetes Java. En el primer paquete “upct.etsit” se encuentra la clase “SpringBootApplication.java”. Es una clase con un método main que es la encargada de arrancar la aplicación.

En el paquete “upct.etsit.config” se encuentran las clases que se encargan de la configuración de la aplicación, en especial de la configuración de la seguridad, y de los CORS Filter. Los CORS (Cross Origin Resource Sharing) son un mecanismo que permite que se puedan solicitar recursos en una página web desde un dominio fuera del dominio que sirvió el primer recurso.

En el paquete “upct.etsit.controller” se encuentran algunos controller del API REST. Al construir servicios web RESTful con Spring, las solicitudes HTTP son manejadas mediante lo que se denomina controllers. Son identificados mediante la anotación @RestController.

```

@RestController
@RequestMapping("/userses")
public class UserController {

    @Autowired
    private UserService userService;

    @RequestMapping(value = "/user", method = RequestMethod.GET)
    public List<Users> listUser() {
        return userService.findAll();
    }
}

```

Figura 34: Ejemplo de una clase controller.

En el paquete “upct.etsit.model” se encuentran todos los objetos entidad con sus constructores, getters and setters.

En el paquete “upct.etsit.repository” se encuentran las interfaces de repositorio del proyecto. Existe una interfaz de repositorio por cada objeto entidad en el paquete “upct.etsit.model”. Los repositorios se utilizan en Spring Data JPA para almacenar datos en una base de datos relacional.

En el paquete “upct.etsit.service” se encuentra

En el paquete “upct.etsit.service.impl”

En el paquete “upct.etsit.utils” se encuentra una clase que ayuda a codificar un String y a comparar dos Strings encriptados con BCrypt.

En el paquete “upct.etsit.view” se encuentran clases necesarias para poder generar el PDF de un curriculum.

En el paquete “upct.etsit.viewModels.Controller” se rellenan los constructores de las entidades que tienen algunas relaciones ManyToMany o ManyToOne para obtener el objeto en JSON.

En el paquete “upct.etsit.viewModels.Models” se crean los constructores de las entidades que tienen relaciones ManyToOne o ManyToMany.

Una vez explicado el contenido de los paquetes de la aplicación, se pasa a la carpeta src/main/resources donde hay dos archivos: application.properties y data.sql.

En el application.properties se han definido propiedades de la Base de Datos, la URL, el usuario, la contraseña, el driver, el esquema etc. El archivo data.sql, es un script de SQL donde se inicializa con valores algunas tablas de la Base de Datos.

4.4.2. Cliente

Para programar la parte del cliente se ha utilizado NodeJS y Angular. El IDE utilizado ha sido el Visual Studio Code. Una vez descargado e instalado NodeJS y VisualStudioCode, para instalar Angular basta con abrir un terminal en VisualStudio y escribir el comando:

```
npm install -g @angular/cli
```

Esto iniciará un proceso de descarga e instalación de los paquetes necesarios para programar con Angular.

Una vez acabada esa parte, para crear un proyecto se pueden utilizar los siguientes comandos:

```
ng new my-first-project
cd my-first-project
ng serve
```

El primer comando crea un nuevo proyecto, el segundo comando entra dentro de la carpeta del nuevo proyecto, y por último, el tercer comando ejecuta el proyecto.

Componentes.

Para crear el primer componente dentro de la aplicación, basta con navegar hasta la carpeta src/app y poner el siguiente comando:

```
ng generate component animals
```

Esto creará una carpeta llamada “animals” dentro de src/app, y dentro de esta carpeta se pueden encontrar los siguientes archivos:

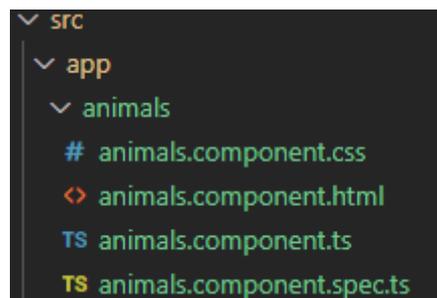


Figura 35: Estructura de la carpeta generada “animals”

Entre los cuatro archivos, se distingue un archivo HTML. Se trata de un lenguaje marcado de hipertexto (HyperText Markup Language) y que interpreta el navegador web para mostrar los sitios o aplicaciones web.

A modo de reseña histórica para resaltar la importancia de HTML, mencionar que hace ya casi 30 años Timothee Berners-Lee lanzó la primera versión de HTML en 1991. Junto a esto, también es fundador de la W3C (World Wide Web Consortium), creador del protocolo HTTP (HyperText Transfer Protocol), de la URL, y del primer navegador web. Se le conoce como el padre de la web.

HTML sigue una estructura para el desarrollo de las páginas web que es la que se muestra en la figura 36:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <title>Mi página web</title>
  </head>
  <body>
    <h1>Hola Mundo</h1>
  </body>
</html>
```

Figura 36: Estructura de un archivo HTML

Ese código mostrará lo siguiente en un navegador:

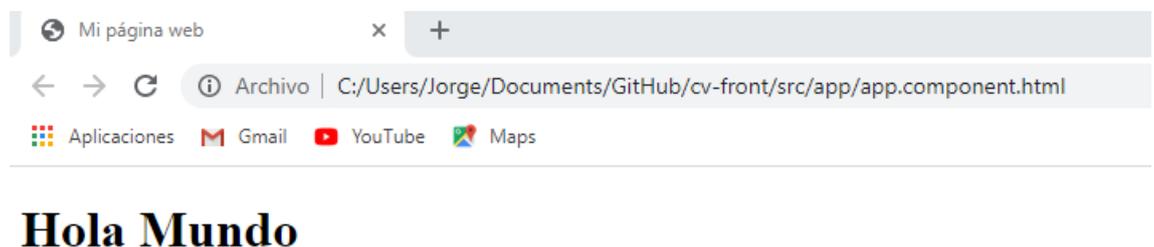


Figura 37: Interpretación de la figura 36 en un navegador

El siguiente archivo que se puede ver, es un archivo tipo CSS. CSS significa *Cascading Style Sheets*. Se utiliza para dar estilos a una página web con la finalidad de que quede mejor estéticamente. Fue desarrollado por W3C en 1996.

CSS tiene una estructura de sintaxis muy simple. Se selecciona un elemento y se escribe lo que se quiere hacer con el mismo.

```
html {  
  background-color: red;  
}  
  
h1 {  
  text-align: center;  
}
```

Figura 38: Estructura de un archivo CSS

El código CSS que se muestra en la figura 38 lo que hace es, en la etiqueta html, establecer el color rojo de fondo, y en la etiqueta h1, alinea el texto para que se muestre en el centro de la página. El resultado puede verse en la figura 39.

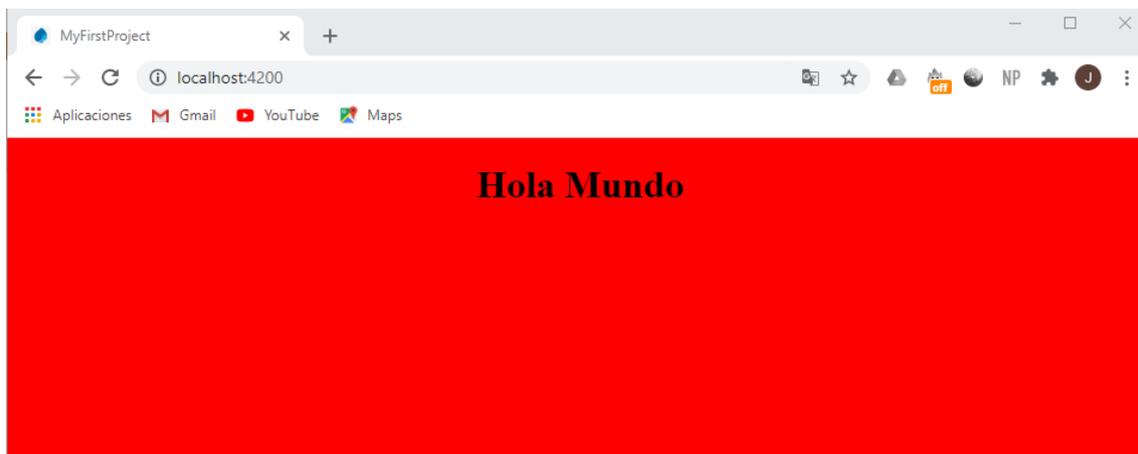


Figura 39: Interpretación de la figura 38 en un navegador

Otro archivo que se puede apreciar es un archivo TypeScript (ts). Se trata de un lenguaje de programación de código abierto desarrollado por Microsoft. Su primera versión fue publicada en 2012. En este proyecto, este archivo permitirá darle funcionalidad a la web, haciendo la conexión con la Base de Datos, recuperando datos, inyectando, modificando o eliminando.

Por último, se encuentra un archivo para pruebas unitarias spec.ts. Al generar un nuevo proyecto, se puede omitir escribiendo el siguiente comando:

```
ng new my-first-project --skip-test
```

Estructura.

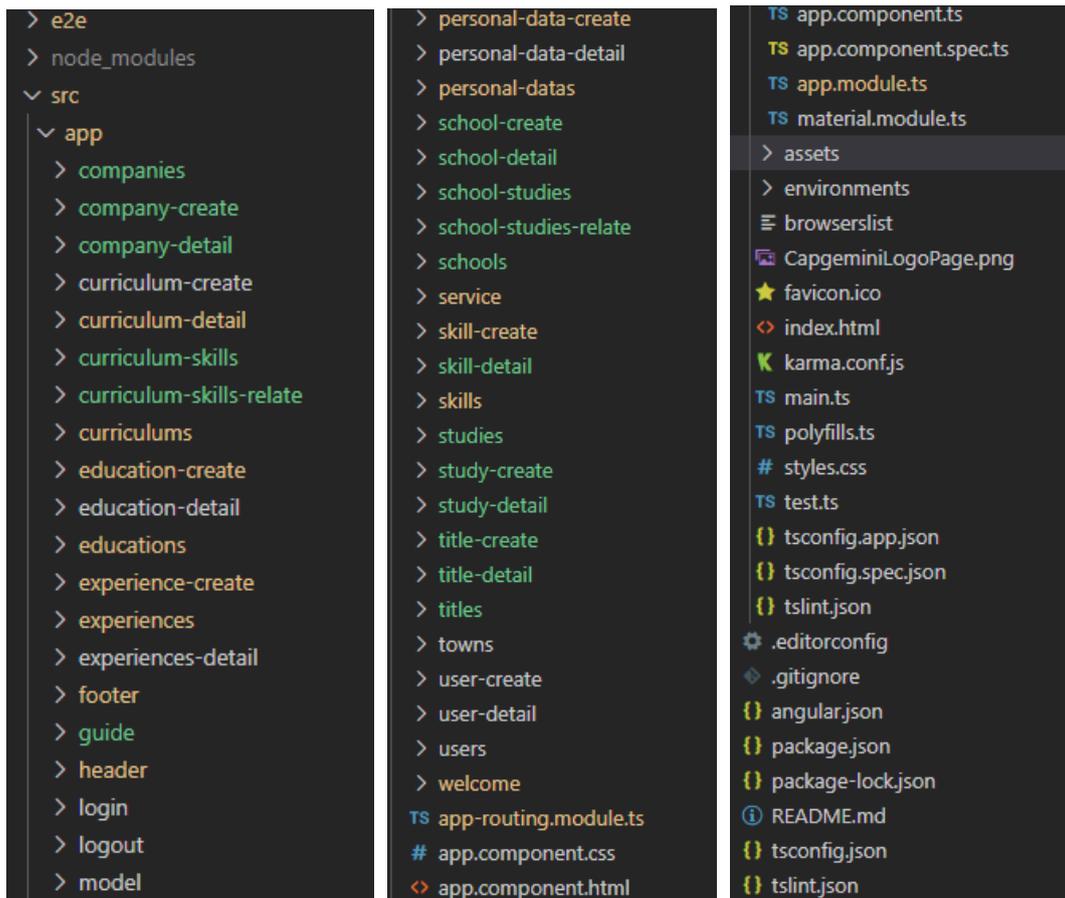


Figura 40: Estructura del frontend.

Como se ve en la figura 40, la estructura que se encuentra del lado del *frontend* para permitir la vista cliente es más extensa que la que se encuentra en la figura 33 correspondiente al *backend*.

El contenido de la primera carpeta “companies” se encarga de listar en una tabla todas las compañías que están registradas en la aplicación. Desde esa página se podría buscar una compañía por su nombre, añadir una compañía, borrarla o editarla. Otras carpetas que tienen la misma funcionalidad son las de curriculums, educations, experiences, personal-datas, schools, skills, studies, titles, towns y users.

El contenido de la carpeta “company-create” tiene la funcionalidad de crear una compañía, es decir, un formulario cuya finalidad es añadir a la Base de Datos el nombre de la compañía. Otras carpetas que tienen una funcionalidad parecida a esta son las carpetas curriculum-create, education-create, experience-create, personal-data-create, school-create, skill-create, study-create, title-créate y user-create.

El contenido de la carpeta “company-details” tiene la funcionalidad de editar el nombre de una compañía. Para acceder a este menú hay que ir a la pestaña “Companies” del menú y posteriormente clickar en el lápiz. Otras carpetas que contienen la misma funcionalidad son curriculum-details, education-details, experience-details, personal-data-details, school-details, skill-details, study-details, title-details y user-details.

Existe también una carpeta llamada “model”. Esta carpeta contiene la definición de modelos de las clases que se pueden usar, sería el equivalente de las entities en Java.

La carpeta services contiene los controladores de los métodos HTTP que se van a utilizar. Por ejemplo, si se quiere crear un usuario, la funcionalidad para procesar la información del método POST para dar de alta en la base de datos al usuario se programa en el “user.service”, así como la funcionalidad para actualizarlo (POST), obtener todos los usuarios (GET) o borrar un usuario (DELETE).

Dentro de la carpeta assets se encontrarán los asset y archivos adicionales para hacer que el proyecto funcione correctamente.

Dentro de la carpeta enviroments se encuentran las variables de entorno.

El archivo main.ts es el archivo inicial del proyecto TypeScript. Aquí se podrán configurar todas las configuraciones globales del mismo.

En el archivo angular.json se puede encontrar información relativa de Angular, así como rutas, versiones, etc.

El archivo package.json es el equivalente al archivo pom.xml de Java, es decir, este archivo contiene la información de la app, como el nombre, la versión, las dependencias que tiene, etc.

4.4.3. GitHub

GitHub[22] es una plataforma para alojar el código de las aplicaciones. La web utiliza el sistema de control de versiones Git, del que ya se ha hablado en el apartado 4.3.2. Además de permitir ver el código y descargar las diferentes versiones del código que se hayan subido, esta plataforma también hace la función de red social, conectando a desarrolladores para que estos puedan colaborar en una aplicación conjunta. Fue comprada por Microsoft en 2018.

GitHub da permiso a los desarrolladores para alojar sus proyectos en repositorios de forma gratuita, pero para esto, deberán ser de código abierto.

Además del sistema de Git, GitHub posee otras herramientas como la de crear una Wiki para cada proyecto donde poner información sobre él.

También dispone de un sistema de seguimiento de problemas. Ofrece una herramienta de revisión de código, de forma que permite dejar anotaciones aparte de mirar el código.

Posee además un IDE que facilita la gestión de repositorios llamado GitHub Desktop, su descarga es gratuita y puede obtenerse de la página oficial de GitHub.

4.4.4. Jenkins

Jenkins [\[23\]](#) nació en 2011, a pesar de que su desarrollo empezó en 2004 en Sun Microsystems.

Se trata de es una herramienta de integración continua. La integración continua consiste en hacer compilaciones y ejecuciones de un proyecto de forma automática y frecuentemente. Una de las ventajas que posee la integración continua es la detección y solución de errores.

Jenkins puede trabajar con Git, y permite ejecutar proyectos basados en Apache Maven entre otros, por lo que la integración en este proyecto resultaba muy necesaria.

Por último, cabe destacar que Jenkins puede extenderse mediante plugins.

Tareas de Jenkins

Se han configurado dos tareas en Jenkins. La primera es una tarea de compilación y despliegue del proyecto *Backend*. Esta tarea simplemente ejecuta el comando “mvn install”, lo que crea un .jar del proyecto y lo guarda en una carpeta. Cuando finaliza, se ejecuta un comando de cmd `java -jar <ruta_del_archivo>` que provocará que la aplicación se despliegue.

La segunda tarea trata de la compilación y despliegue del *frontend*. En primer lugar, se ejecutará `npm install`, que instalará las dependencias del proyecto Angular, y después se ejecutará `npm start`, para desplegar el proyecto.

Capítulo 5 - Conclusiones y trabajo futuro

En este capítulo se expondrán las conclusiones, así como el trabajo futuro.

5.1. Conclusiones

Finalmente, y en concordancia con los objetivos de este proyecto, se ha podido llevar a cabo la creación de un API REST con Spring Boot y consumida con Angular7 que va a permitir la gestión de curriculums para la empresa Capgemini de una manera mucho más funcional a la hora de asignar sus proyectos debido a su filtro por skills.

Como se ha mencionado anteriormente, la idea de este proyecto surgió, , en primer lugar, de la necesidad que tenía la empresa de tener un software que pudiera ayudar en la gestión de sus empleados y/o candidatos a la entrada de sus proyectos y ha resultado, y en segundo lugar, de mi inquietud personal por abordar un gran reto tanto en el ámbito personal como en el profesional, ya que el desarrollo se ha efectuado desde cero y teniendo conocimientos muy básicos de ciertas tecnologías aplicadas.

El desarrollo de este proyecto me ha permitido profundizar en el desarrollo de aplicaciones *Backend*, además del aprendizaje en conceptos y ramas sobre desarrollo de *Fronted*, entre otras. Por lo que me gustaría agradecer a la empresa Capgemini el haberme dado la oportunidad de realizar este proyecto con ellos, ya que me ha ayudado a enfocar mi carrera profesional.

Como resultado, he sido capaz de hacer un desarrollo FullStack (desarrollo de Backend y desarrollo de Frontend) de una aplicación con la que el equipo de Recursos Humanos de Capgemini será capaz de gestionar y saber en todo momento que perfiles de programadores y programadoras dispone de cara a la recepción de nuevos proyectos.

5.2. Trabajo futuro

A continuación, se proponen algunas líneas de trabajo que podrían abordarse para ampliar o mejorar este proyecto:

- Implementación de clasificación inteligente. Se podría implementar mediante TensorFlow una clasificación inteligente de los perfiles de programación en función de las skills que aparezcan en el curriculum. Los perfiles podrían ser: desarrollador de *backend* (si controla tecnologías de Java, Maven, SOAP, REST, Apache CXF, etc.), desarrollador de *frontend* (si controla tecnologías de HTML, TypeScript, JavaScript, CSS, Angular, Reactjs, etc.), desarrollador Fullstack (si controla tecnologías *backend, frontend* etc.).
- Añadir listado de proyectos. Puede añadirse una funcionalidad con un registro de todos los proyectos en los que se ha participado dentro de la empresa.
- Modificar el conjunto de datos "Educations" para poder añadir estudios que no sean universitarios, como estudios de Formación Profesional, curso emitidos por entidades certificadoras (como Cisco, por ejemplo), o incluso cursos online emitidos por entidades de prestigio.

Capítulo 6 – ANEXOS

ANEXO I: Manual de usuario.

En este anexo se adjunta una guía rápida de uso de la aplicación.

En la figura 42 se muestra la página de inicio que se ve al entrar en la aplicación:

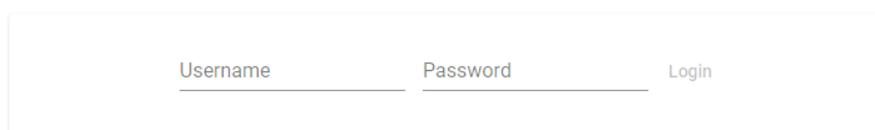


Ilustración de la pantalla de login. Muestra tres campos de entrada: 'Username', 'Password' y un botón 'Login'.

Figura 41: Ilustración que muestra la pantalla de login

Para acceder con permisos de usuario se utilizarán las siguientes credenciales:

Usuario: user

Contraseña: user

Accediendo de esta manera a la aplicación, sólo se podrá consultar y editar los datos de los curriculums que ya hay registrados, registrar un currículum nuevo, y hacer una búsqueda por Skills para nuevos proyectos.

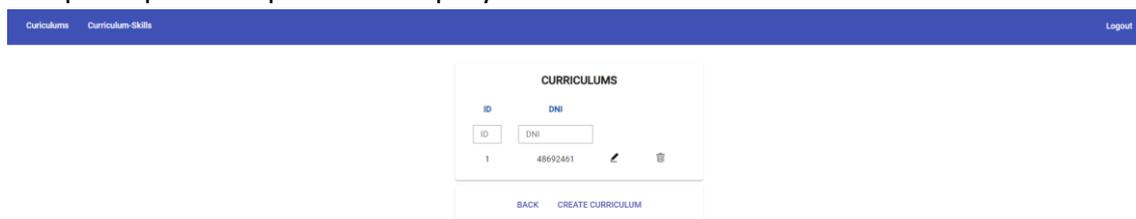


Ilustración de la página de inicio de la aplicación. Muestra un menú superior con 'Curriculum' y 'Curriculum-Skills', un botón 'Logout', y un panel central con el título 'CURRICULUMS' que contiene una tabla con una fila de datos (ID: 1, DNI: 48692461) y botones 'BACK' y 'CREATE CURRICULUM'.

Figura 42: Ilustración que muestra la página de inicio

Creación de un curriculum

Se sitúa el indicador en la pestaña "Curriculum" donde debería aparecer un listado con los curriculum que hay registrados y se pulsa sobre "CREATE CURRICULUM". En primer lugar, se introduce el número de DNI sin la letra.

DNI *
73840911

Next ➤

Figura 43: Ilustración del inicio del formulario de registro de curriculum

A continuación, se deberá rellenar los datos personales, sin olvidar seleccionar en el desplegable del “DNI” el DNI de la persona de la cual se esté dando de alta el curriculum. Una vez terminado, se pulsa en “NEXT TO EDUCATIONS”.

Name *
Andrés

Last Name *
Diaz Lopez

Address *
Zorrilla, 34

Zip Code *
30006

Telephone Number *
657091283

Email *
andres.diaz@gmail.com

Country
España ▼

Town
Murcia ▼

DNI
73840911 ▼

NEXT TO EDUCATIONS

Figura 44: Ilustración que muestra la pestaña de datos personales del formulario de registro de curriculum

En esta pestaña se tendrá que rellenar los estudios (Grado o Máster). Para guardar se pulsa sobre ADD EDUCATIONS y volverá a dirigir a la página. Si se quiere añadir otra educación se repite el proceso, de lo contrario se pulsa “NEXT TO EXPERIENCES”.

School
Universidad Politécnica de Madrid

Field Of Study
Grado en Ingeniería de Computadores

From Year
2010

To Year
2014

DNI
73840911

[BACK TO PERSONAL DATA](#) [ADD EDUCATION](#) [NEXT TO EXPERIENCES](#)

Figura 45: Ilustración de la pestaña “añadir educación” del formulario de registro de curriculum

En esta pestaña se encuentra la misma mecánica que la ventana anterior: para guardar hay que pulsar “ADD EXPERIENCE”, y para avanzar “GO TO SKILLS”.

Company
Capgemini

Title
Estudiante en Prácticas

From Year
2014

To Year
2014

From Month
Marzo

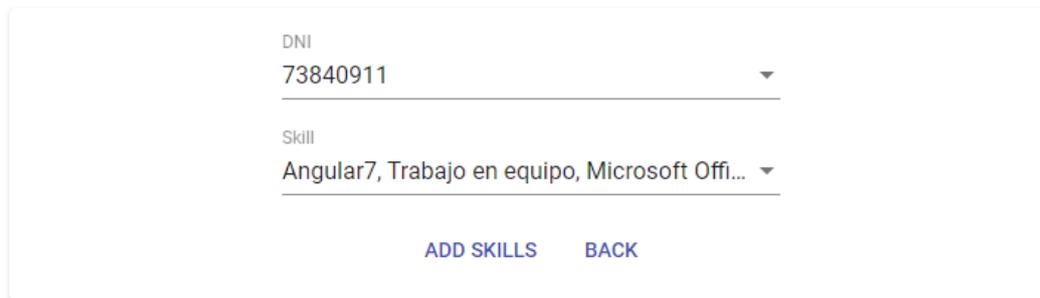
To Month
Junio

DNI
73840911

[BACK TO EDUCATIONS](#) [ADD EXPERIENCE](#) [GO TO SKILLS](#)

Figura 46: Ilustración de la pestaña “añadir experiencia” del formulario de registro de curriculum

En esta pestaña se seleccionan las skills y se procede a “ADD SKILLS” para guardar. Tras este punto, ya se habrá terminado el registro del currículum y ya se podría ir a la ventana “CURRICULUMS” para acceder a él.



The image shows a screenshot of a web form titled "ADD SKILLS". It contains two dropdown menus. The first is labeled "DNI" and has the value "73840911" selected. The second is labeled "Skill" and has the value "Angular7, Trabajo en equipo, Microsoft Offi..." selected. Below the dropdowns are two buttons: "ADD SKILLS" and "BACK".

Figura 47: Ilustración de la pestaña “añadir skill” del formulario de registro de curriculum

Para acceder al currículum basta con pulsar sobre el lápiz que hay al lado de la columna “DNI”. Se abrirá una pestaña donde consultar toda la información referente al currículum.

id

3

PERSONAL DATA

Name

Diaz Lopez, Andrés

Address

Zorrilla, 34 - 30006, Murcia, España

Contact

657091283 - andres.diaz@gmail.com

EDUCATIONS

- Grado en Ingeniería de Computadores
Universidad Politécnica de Madrid
2010 - 2014

EXPERIENCE

- Estudiante en Prácticas en Cppgemini
From Marzo 2014 to Junio 2014
- Junior Software Developer en Cppgemini
From Julio 2014 to Diciembre 2019
- Senior Software Developer en Cppgemini
From Diciembre 2019 to Mayo 2020

SKILLS

- Angular7
- Trabajo en equipo
- Microsoft Office
- Microsoft Word
- Gestión de Proyectos
- Investigación y Desarrollo
- Redes Sociales
- Medios Sociales
- Microsoft Excel
- HTML
- PHP
- Liderazgo
- JavaScript
- Desarrollo Web
- CSS

[BACK](#) [GENERATE PDF](#)

Figura 48: Ilustración del resumen de un curriculum

Si se pulsa sobre “GENERATE PDF”, se generará un PDF con el currículum que tendrá como título el número de DNI.

En la pestaña “Curriculum-Skill” se encuentra un cuadro con todas las skills que hay registradas en la BBDD y los usuarios a las que están asignadas.

SKILLS

DNI	Nombre	Apellidos	Skill		
48692461	Jorge	Zapata Alburquerque	Java	✎	🗑
48692461	Jorge	Zapata Alburquerque	Angular7	✎	🗑
48692461	Jorge	Zapata Alburquerque	SOAP	✎	🗑
48692461	Jorge	Zapata Alburquerque	Web Services	✎	🗑
48692461	Jorge	Zapata Alburquerque	Trabajo en equipo	✎	🗑
48692461	Jorge	Zapata Alburquerque	Microsoft Office	✎	🗑
48692461	Jorge	Zapata Alburquerque	Microsoft Word	✎	🗑
48692461	Jorge	Zapata Alburquerque	Gestión de Proyectos	✎	🗑
48692461	Jorge	Zapata Alburquerque	Investigación y Desarrollo	✎	🗑
48692461	Jorge	Zapata Alburquerque	Redes Sociales	✎	🗑

←1234→

BACK

Figura 49: Ilustración que representa el resumen de todos los empleados con las skills que posee cada uno

También se puede hacer un filtrado y ver que curriculums tienen una Skill, como puede ser Angular7.

SKILLS

DNI	Nombre	Apellidos	Skill		
48692461	Jorge	Zapata Alburquerque	Angular7	✎	🗑
73840911	Andrés	Diaz Lopez	Angular7	✎	🗑

BACK

Figura 50: Ilustración que representa el filtrado de la Skill "Angular 7"

ANEXO II: Manual de administrador.

Para acceder con permisos de administrador se deberán utilizar las siguientes credenciales:

usuario: admin

contraseña: admin.

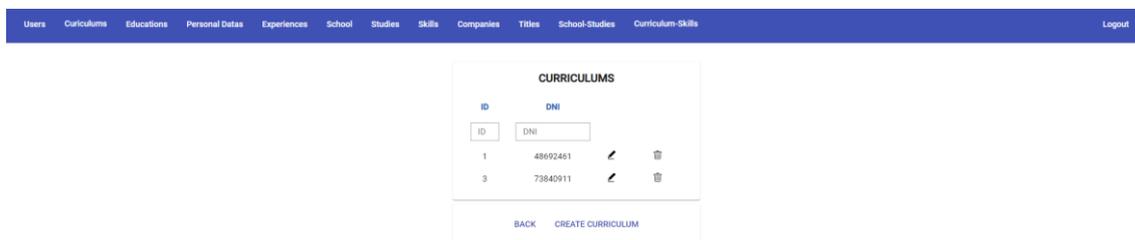


Figura 51: Ilustración que muestra la página de inicio de la app.

Como se puede observar, si se loguea un usuario con roles de administrador en el menú aparecen más opciones.

Con un perfil de administrador se tendrá una total disponibilidad de la aplicación, pudiendo crear, editar y borrar Compañías, Estudios, Universidades, Usuarios, etc.

En el menú “Educations” se puede ver un listado de todas las educaciones que tiene la aplicación.



Figura 52: Ilustración que muestra las Educaciones que hay creadas en la aplicación

Si se pulsa sobre “CREATE EDUCATION” se redireccionará a una ventana similar a la de registro de curriculum.

En la pestaña SCHOOLS se puede observar que tiene una particularidad diferente, y es que aparece un botón RELATE STUDIES. Si se pulsa, se abrirá una ventana donde poder relacionar los estudios que se cursan, con una universidad, en caso de que una universidad modificase sus estudios.

SCHOOLS

ID	School		
1	CUNEF Universidad		
2	ESIC Universidad		
3	Saint Louis University Madrid Campus		
4	Universidad a Distancia de Madrid		
5	Universidad Abad Oliva CEU		
6	Universitat Oberta de Catalunya		
7	Universidad Alfonso X el Sabio		
8	Universidad Antonio de Nebrija		
9	Universidad Autónoma de Barcelona		
10	Universidad Autónoma de Madrid		

«<1234->»

[BACK](#) [ADD SCHOOL](#) [RELATE STUDIES](#)

Figura 53: Ilustración que muestra todas las universidades añadidas en la aplicación.



Figura 54: Ilustración que muestra el desplegable de Grados disponibles para añadirlos o modificarlos en una universidad

ANEXO III: Instalación y configuración de Maven para Eclipse.

En primer lugar, se procede a instalar Maven en Windows, descargándolo desde su sitio oficial <https://maven.apache.org/download.cgi>

Una vez descargado y descomprimido en el disco local C, se procede a crear una variable de entorno.

Para ello: Inicio -> Equipo -> Click derecho -> Propiedades

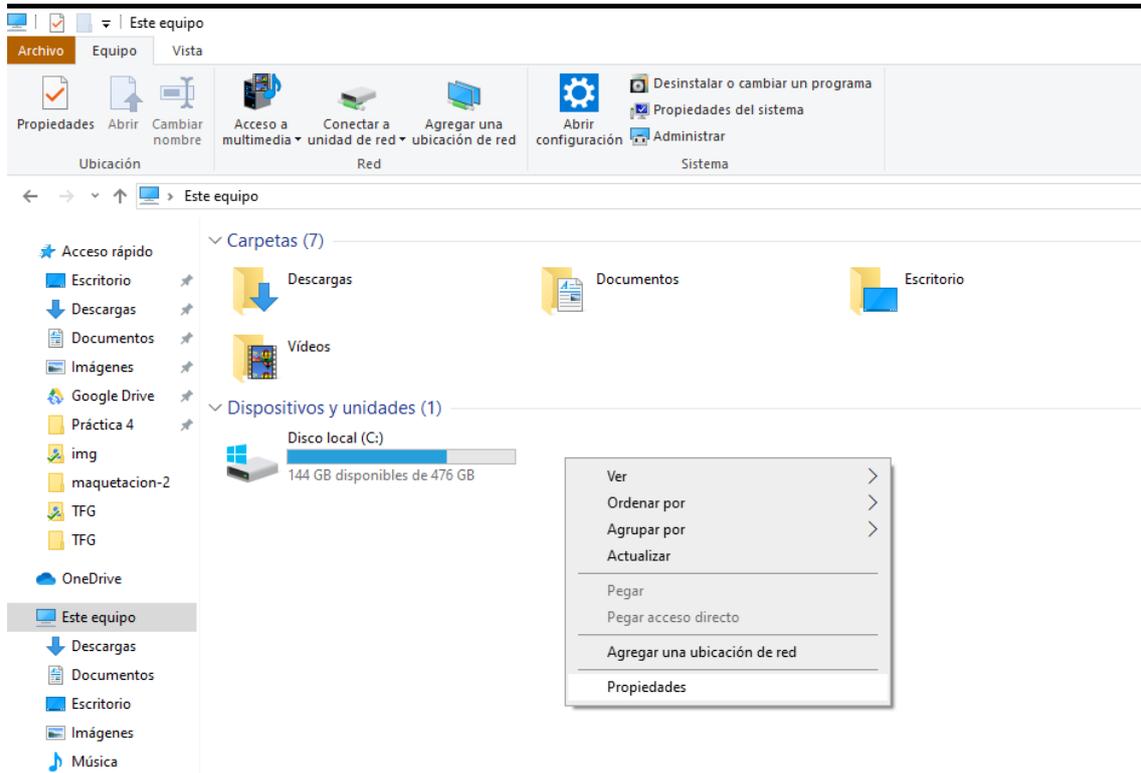
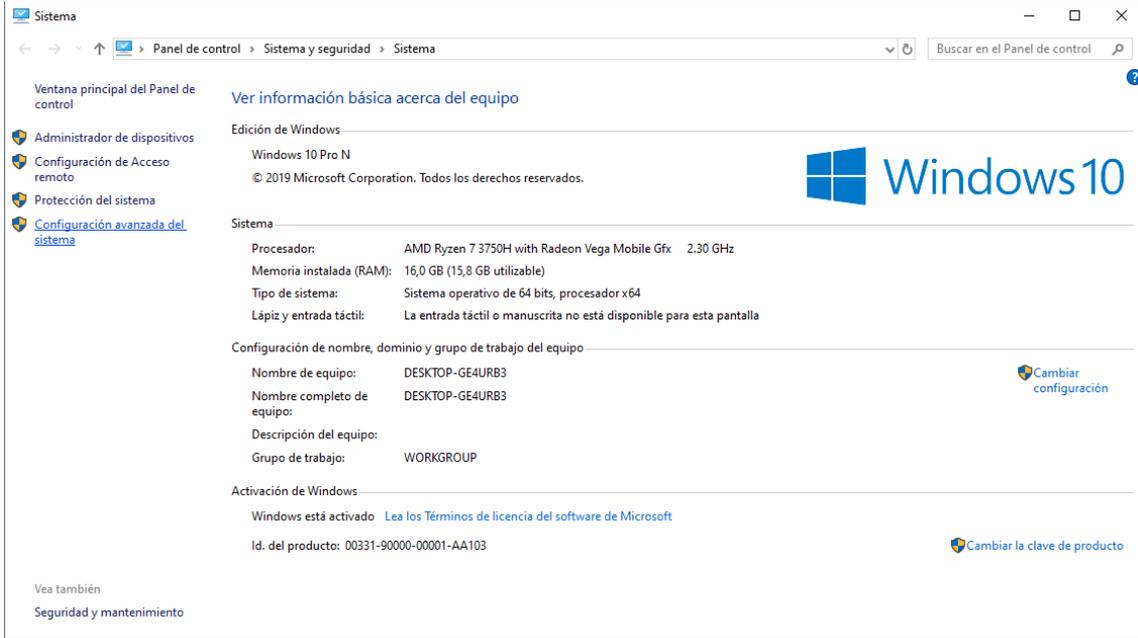


Figura 55: Ilustración que representa el contenido “Este equipo”.

Se dirige a “Configuración avanzada del sistema”



The screenshot shows the Windows 10 'System' settings window. The title bar reads 'Sistema'. The breadcrumb navigation is 'Panel de control > Sistema y seguridad > Sistema'. The left sidebar contains links to 'Administrador de dispositivos', 'Configuración de Acceso remoto', 'Protección del sistema', and 'Configuración avanzada del sistema'. The main content area is titled 'Ver información básica acerca del equipo' and features the Windows 10 logo. It is divided into several sections: 'Edición de Windows' (Windows 10 Pro N, © 2019 Microsoft Corporation), 'Sistema' (Processor: AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz, Memory: 16.0 GB, System type: 64-bit), 'Configuración de nombre, dominio y grupo de trabajo del equipo' (Name: DESKTOP-GE4URB3, Group: WORKGROUP), and 'Activación de Windows' (Windows is activated, Product ID: 00331-90000-00001-AA103). A 'Vea también' section at the bottom left points to 'Seguridad y mantenimiento'.

Ventana principal del Panel de control

Panel de control > Sistema y seguridad > Sistema

Ver información básica acerca del equipo

Edición de Windows

Windows 10 Pro N
© 2019 Microsoft Corporation. Todos los derechos reservados.

Sistema

Procesador: AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx 2.30 GHz
Memoria instalada (RAM): 16,0 GB (15,8 GB utilizable)
Tipo de sistema: Sistema operativo de 64 bits, procesador x64
Lápiz y entrada táctil: La entrada táctil o manuscrita no está disponible para esta pantalla

Configuración de nombre, dominio y grupo de trabajo del equipo

Nombre de equipo: DESKTOP-GE4URB3
Nombre completo de equipo: DESKTOP-GE4URB3
Descripción del equipo:
Grupo de trabajo: WORKGROUP

Activación de Windows

Windows está activado [Lea los Términos de licencia del software de Microsoft](#)
Id. del producto: 00331-90000-00001-AA103

Vea también
Seguridad y mantenimiento

Figura 56: ilustración que representa el contenido de “Sistema”.

Aparecerá una ventana y se pulsa “Variables de entorno”.

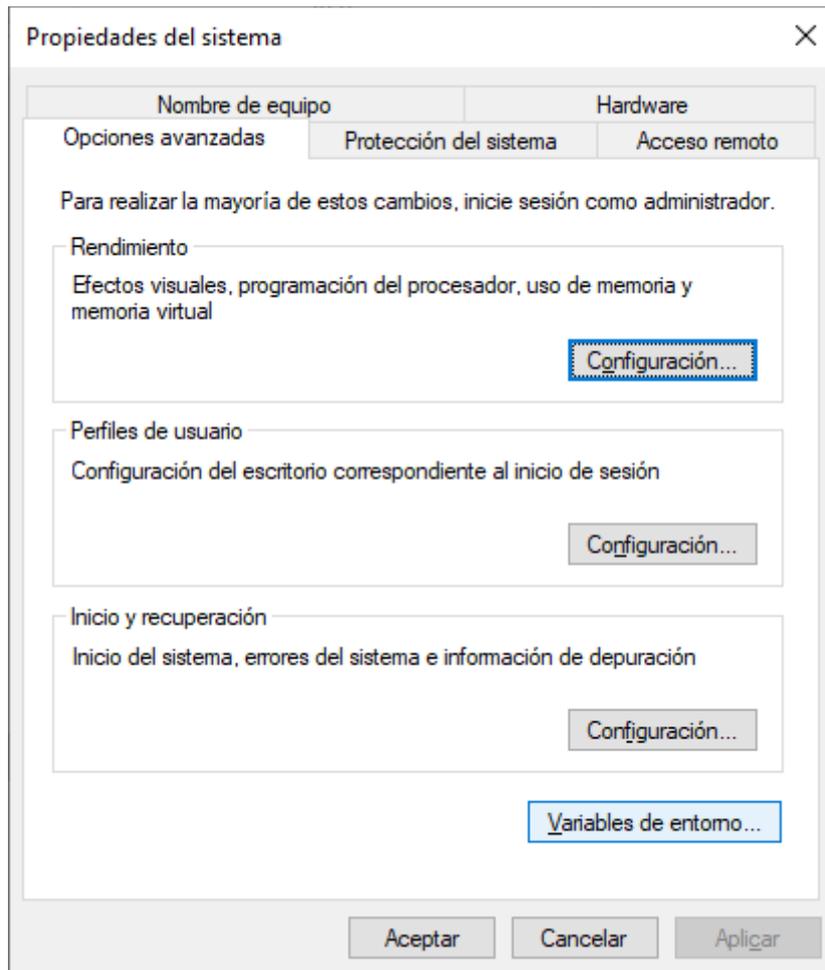


Figura 57: Figura que representa el cuadro de “Configuración avanzada”.

Aparecerá la ventana de variables de entorno, se pulsa “nueva” para añadir una nueva variable.

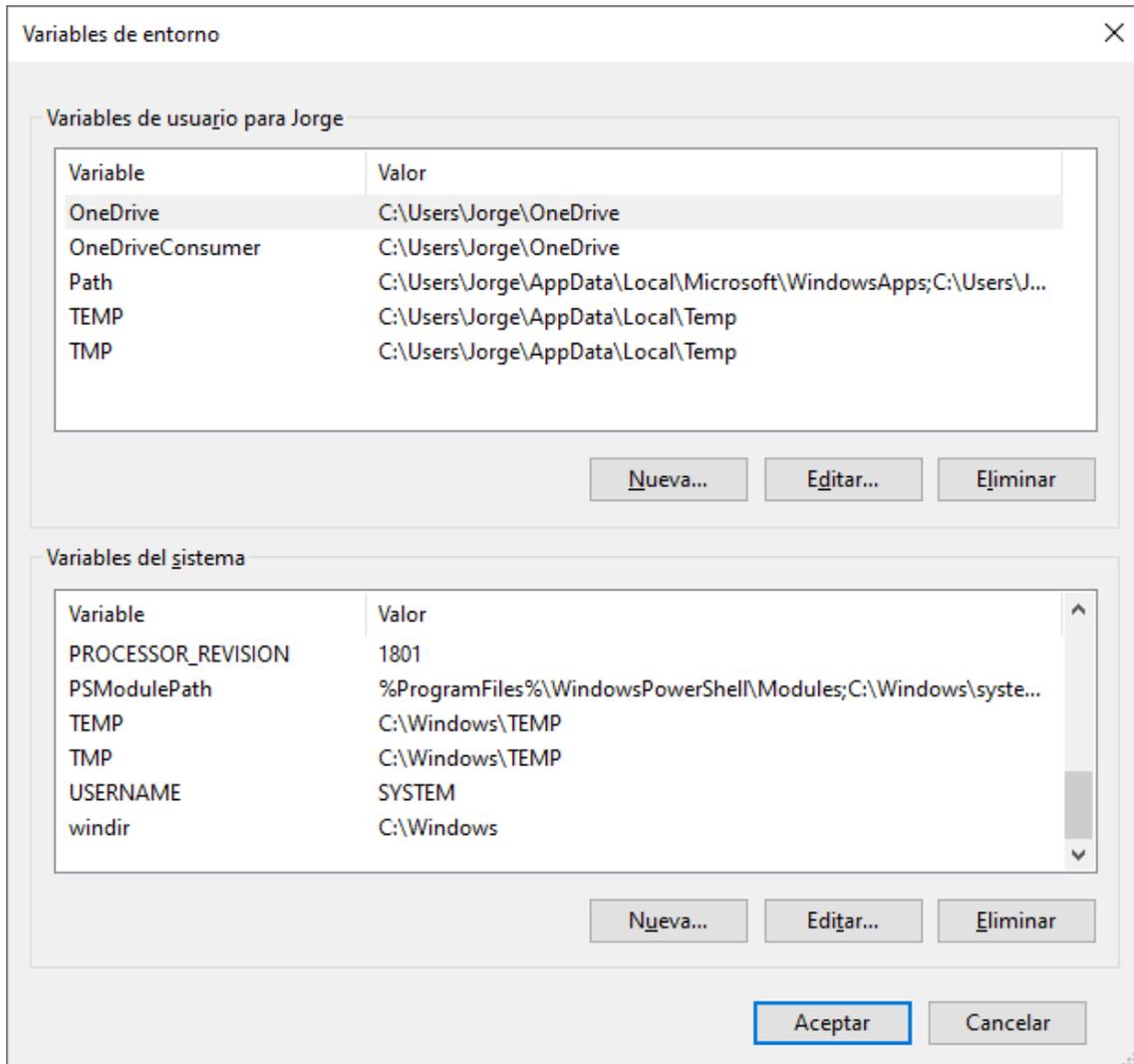


Figura 58: Ilustración que representa la pantalla de variables de entorno

Emergerá un cuadro donde se ingresará el nombre de la variable, a la cual se la llamará M2_HOME, y la ruta donde se encuentra la carpeta de Maven.

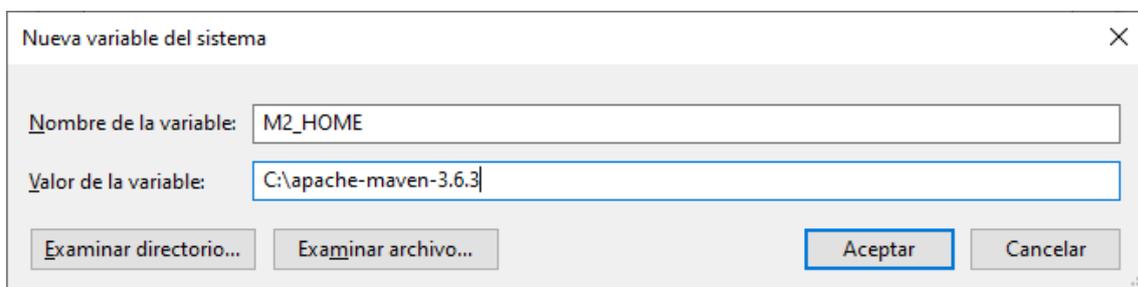


Figura 59: Ilustración que representa el cuadro de nueva variable del sistema.

Lo siguiente es editar la variable de entorno Path para que Windows reconozca la variable M2_HOME. Para ello, en el apartado “Variable del sistema” se busca la variable “Path” y se pulsa en Editar.

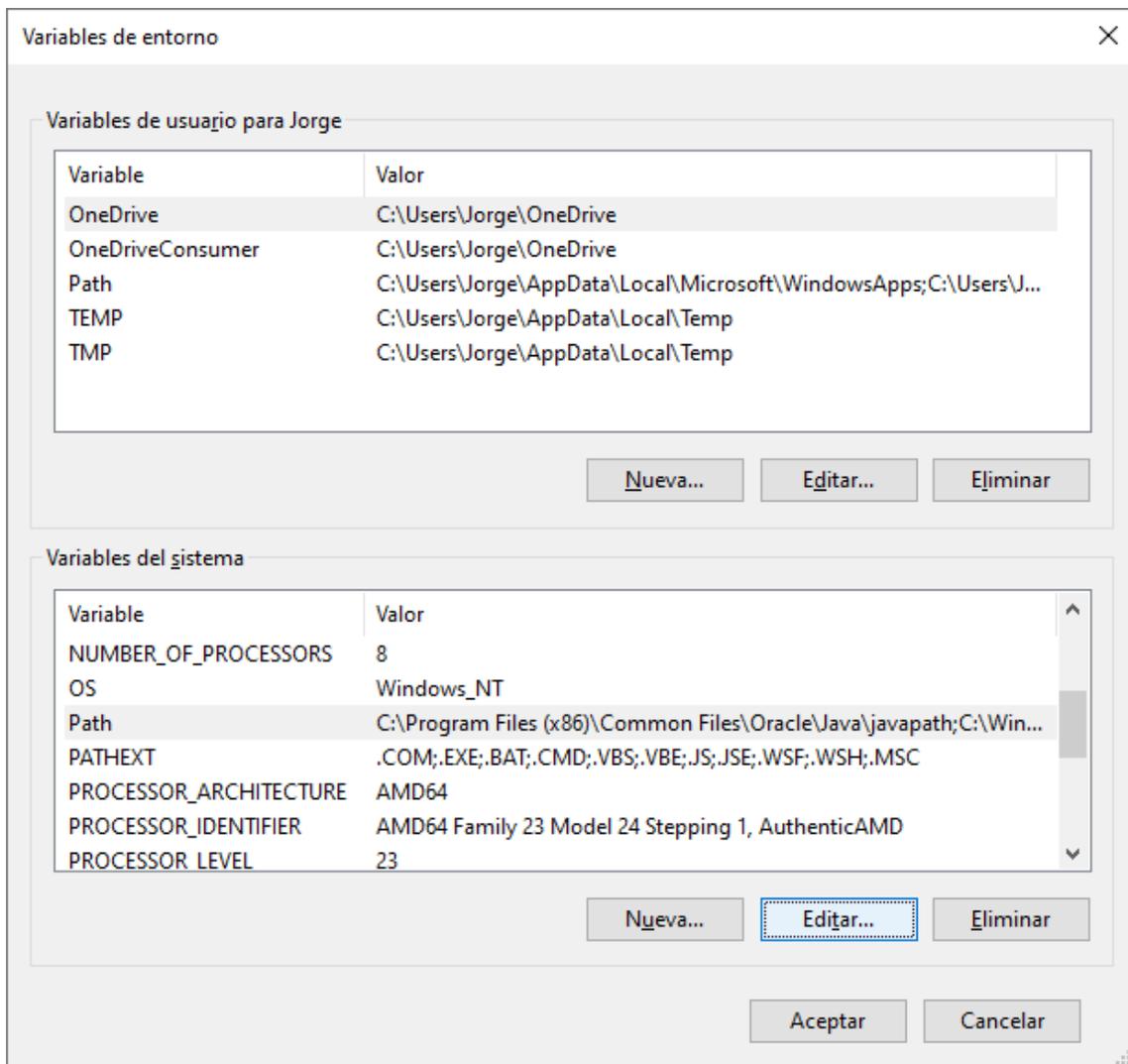


Figura 60: Ilustración que representa la pantalla de variables de entorno.

En este paso únicamente hay que pulsar en “nueva” y escribir la ruta donde está la carpeta bin de Maven.

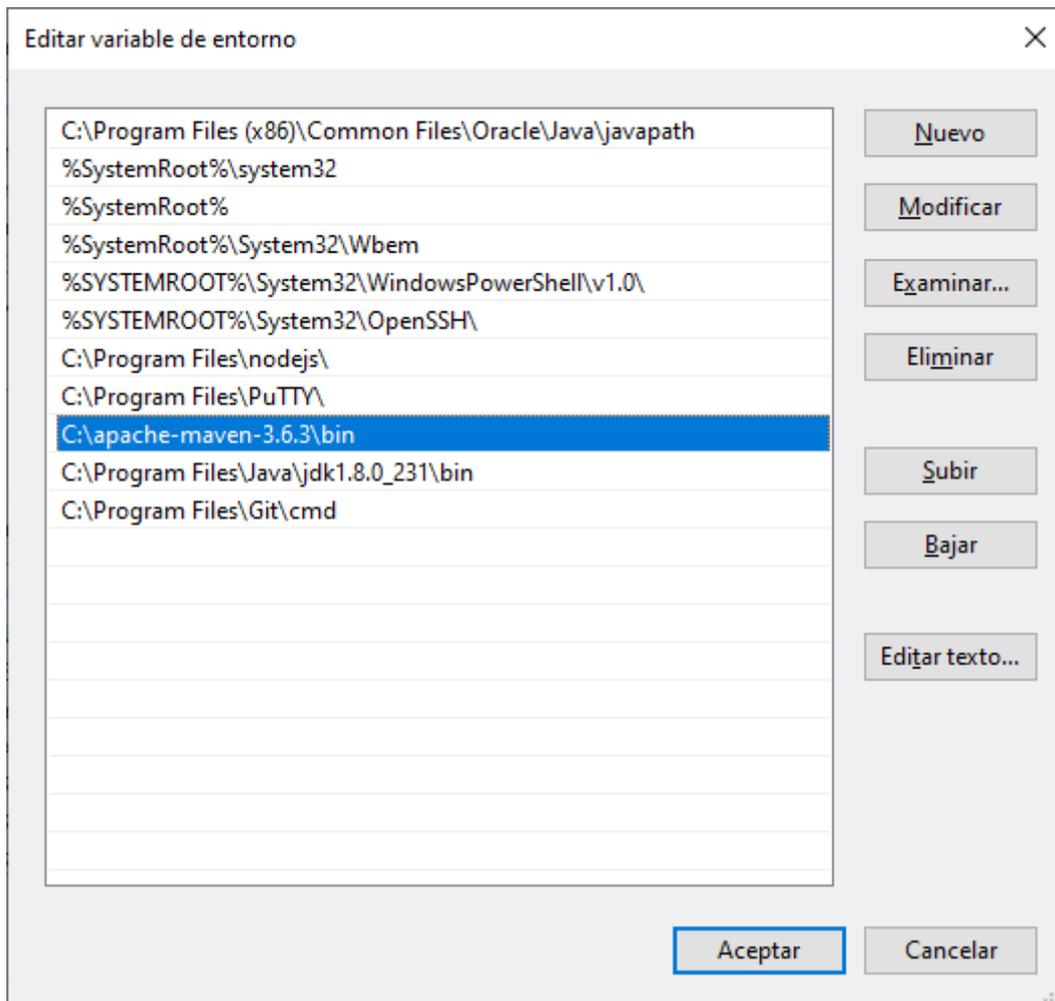
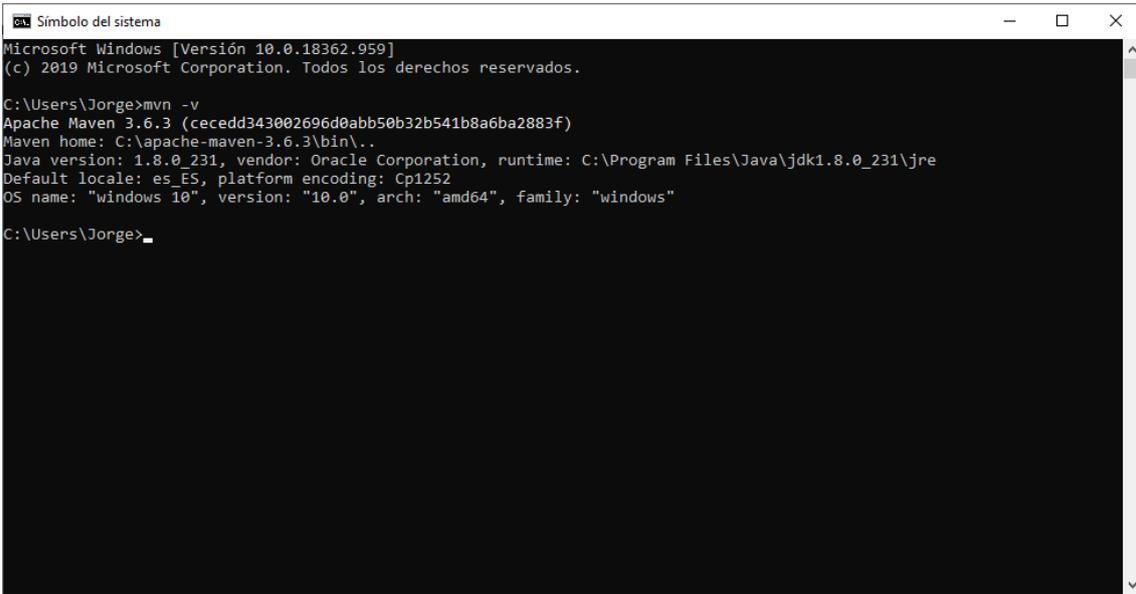


Figura 61: Ilustración que representa la pantalla de edición de una variable de entorno

Ahora ya estaría instalado Maven en el ordenador. Para comprobarlo, se abre un terminal cmd de Windows donde se escribe `mvn -v`, y debería aparecer lo siguiente:



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.959]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jorge>mvn -v
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: C:\apache-maven-3.6.3\bin\..
Java version: 1.8.0_231, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_231\jre
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\Jorge>
```

Figura 62: Ilustración de la Consola de Windows

Ahora se procede a instalar Maven en Eclipse. Para ello, se va al menú de eclipse: “Help -> Install New Software”

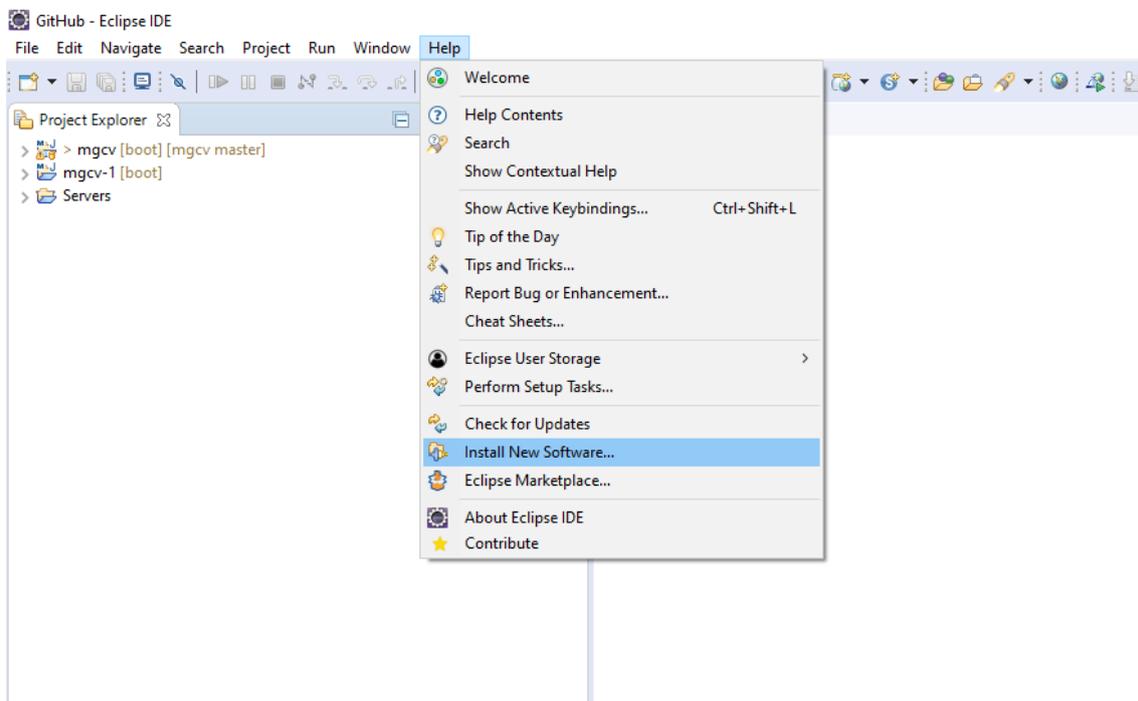


Figura 63: Ilustración que representa donde hay que acceder para instalar un “new software” en Eclipse

En el desplegable “Work with” se selecciona la opción “All Available Sites”.

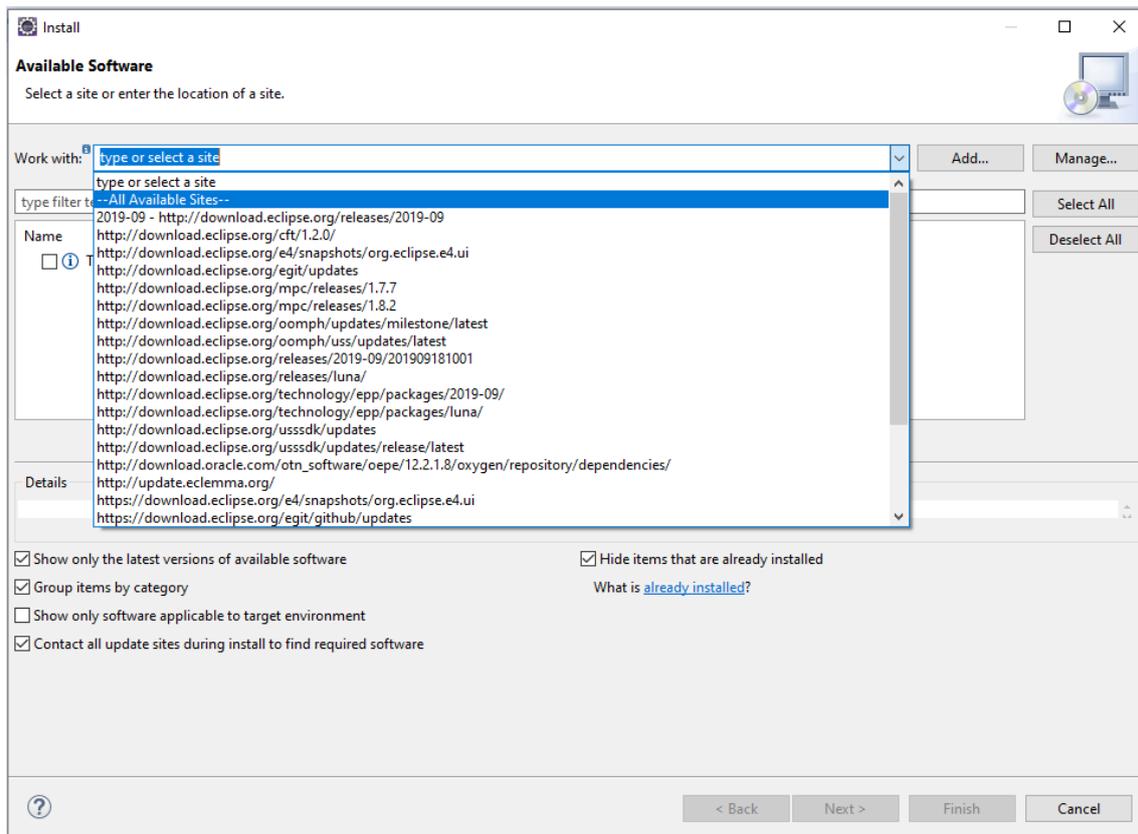


Figura 64: Ilustración que representa los sitios que tiene Eclipse disponibles para buscar software en ellos.

Cuando termine de cargar, se elegirá "Collaboration -> m2e - Maven Integration for Eclipse". Se pulsa Next.

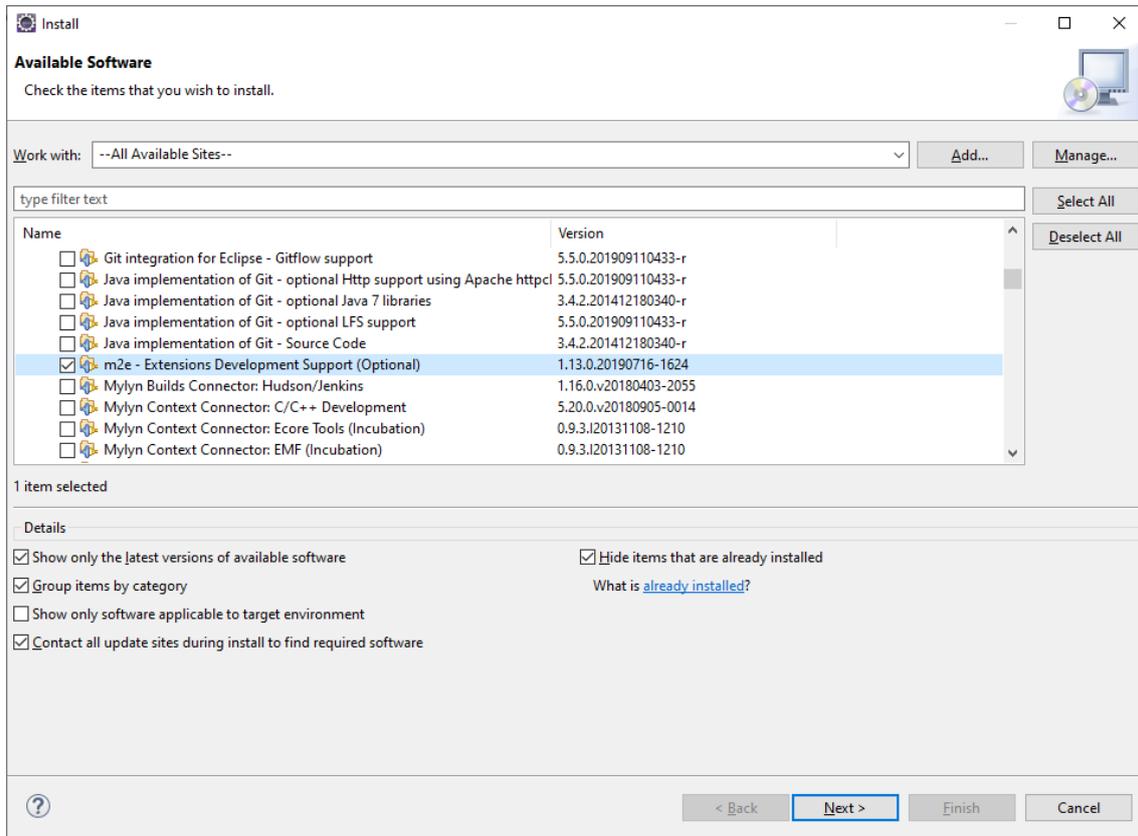


Figura 65: Ilustración que representa una lista de softwares instalados y disponibles para instalar.

Una vez descargado el plugin, pedirá permiso para instalarlo y a continuación permiso para reiniciar Eclipse, para que así se hagan efectivos los cambios.

Una vez haya iniciado Eclipse, se puede comprobar la correcta instalación pulsando en la vista “package Explorer” sobre cualquier proyecto, pulsando el botón derecho y comprobando el menú, debería aparecer la opción “Configure -> Convert to Maven Project”.

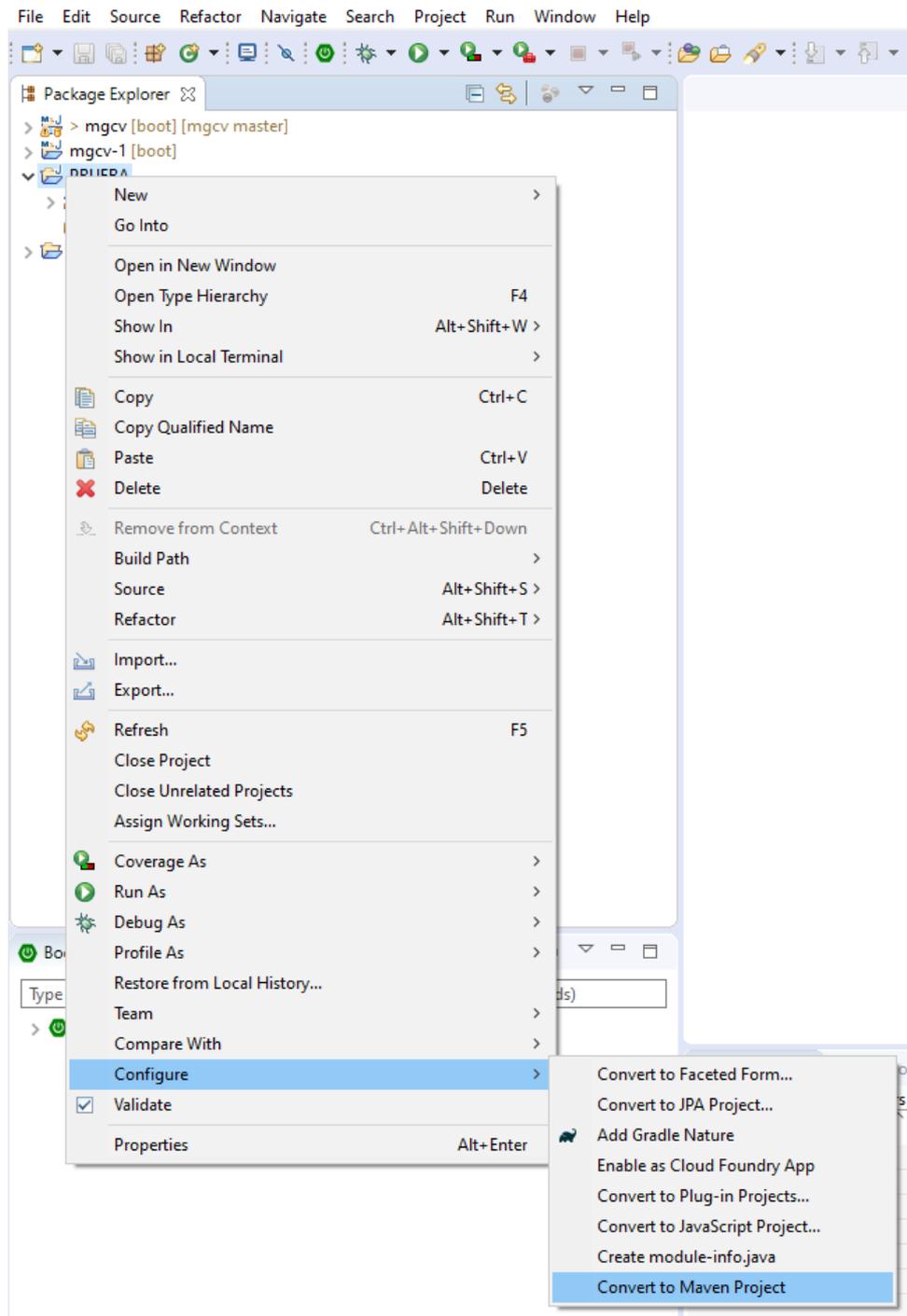


Figura 66: Ilustración que representa los pasos para crear un proyecto Maven

ANEXO IV: Bibliografía

- [1] <https://www.capgemini.com/es-es/>
- [2] <https://www.youtube.com/channel/UC4CEqh4d3-6RcyyJxhFCMGg>
- [3] <https://platzi.com/blog/que-es-frontend-y-backend/>
- [4] www.java.com
- [5] <https://docs.microsoft.com/es-es/dotnet/csharp/>
- [6] <https://spring.io/>
- [7] <https://www.djangoproject.com/>
- [8] <https://maven.apache.org/>
- [9] <https://gradle.org/>
- [10] <https://www.postgresql.org/>
- [11] <https://www.mysql.com/>
- [12] <https://angular.io/>
- [13] <https://es.reactjs.org/>
- [14] <https://www.eclipse.org/ide/>
- [15] <https://code.visualstudio.com/>
- [16] <https://nodejs.org/es/>
- [17] <https://www.pgadmin.org/>
- [18] <https://material.angular.io/>
- [19] <https://kanbantool.com/es/metodologia-kanban>
- [20] <https://trello.com/es>
- [21] <https://git-scm.com/>
- [22] <https://github.com/features>
- [23] <https://www.jenkins.io/>